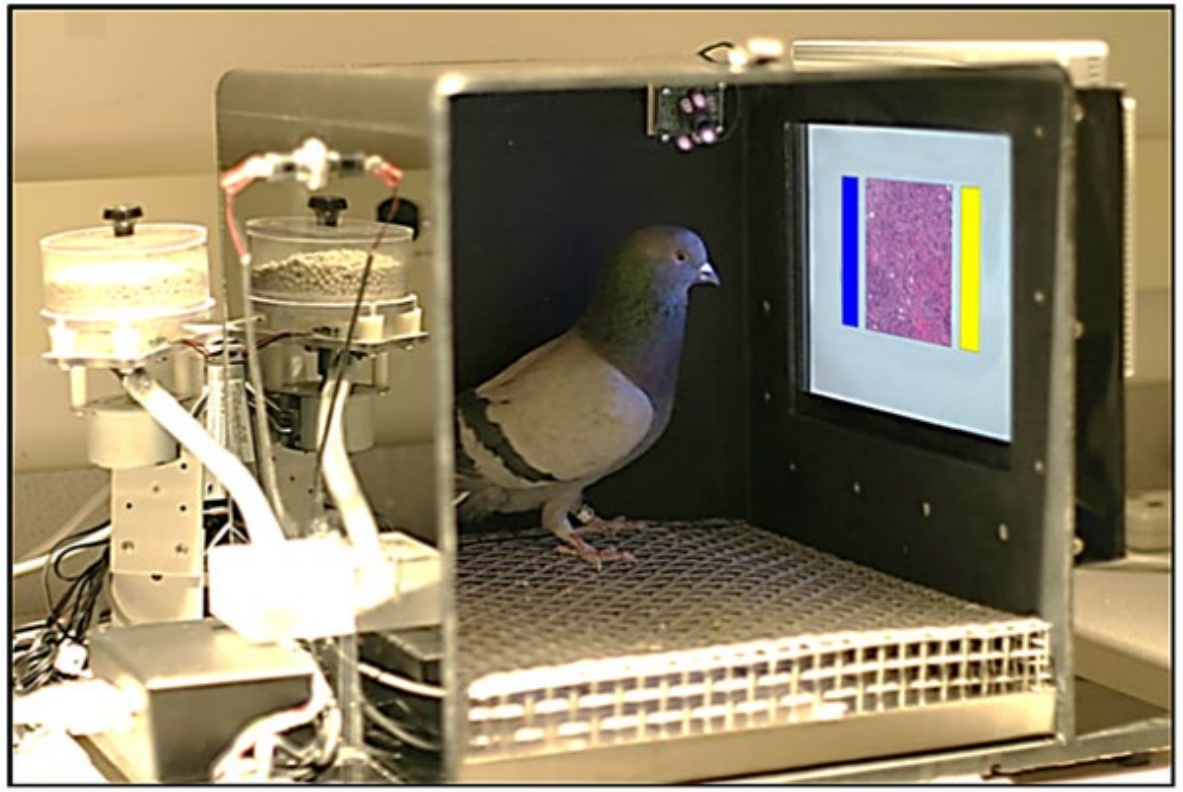
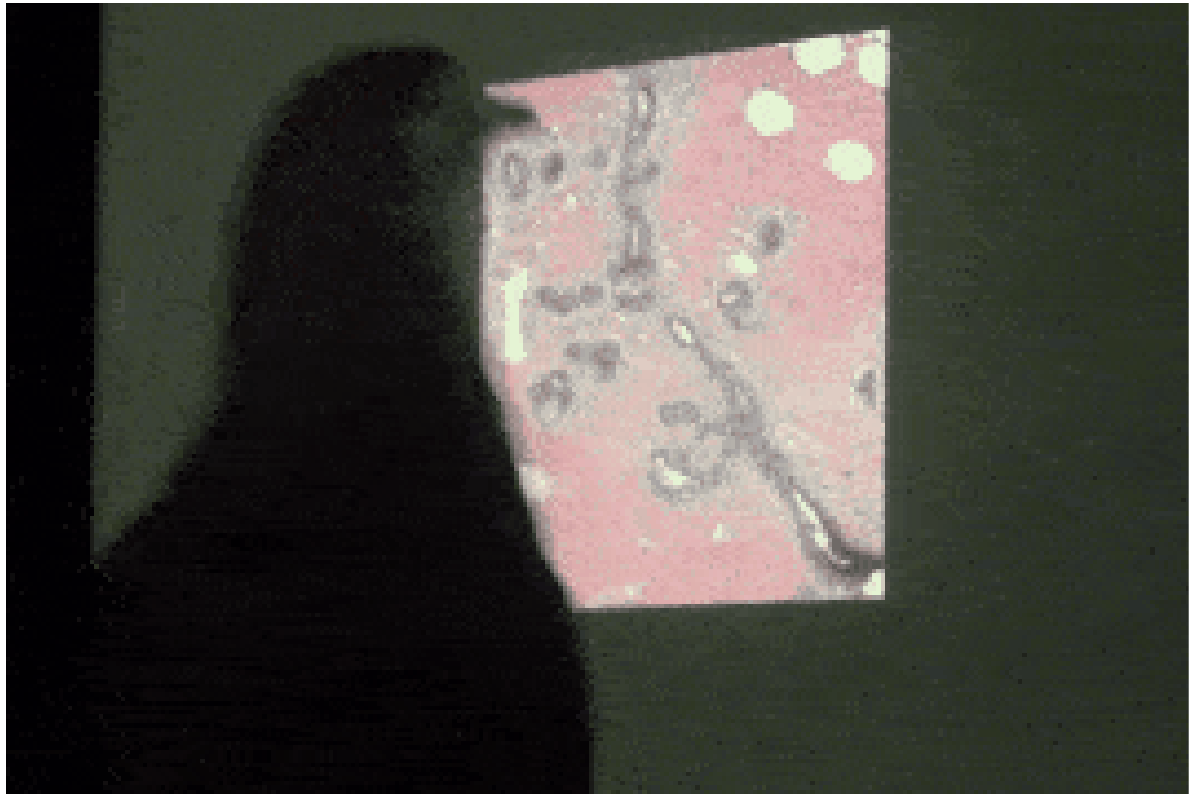
Breast Cancer Classification in Digital Pathology using Python & Deep Learning

PYCON HK 2018

Scotty Kwok

pigeon
The ~~python~~ “deep learning” environment





Levenson RM, Krupinski EA, Navarro VM, Wasserman EA (2015) Pigeons (Columba livia) as Trainable Observers of Pathology and Radiology Breast Cancer Images. PLOS ONE 10(11): e0141357. https://doi.org/10.1371/journal.pone.0141357
https://journals.plos.org/plosone/article?id=10.1371/journal.pone.0141357

PLOS ONE

After two weeks of training … the pigeons reached 85% accuracy.

They successfully identified cancerous tissue from novel images they had not seen before

By pooling a group decision from 4 pigeons, it reached 99% accuracy !

#About me

I am a developer/researcher/co-founder/ IT support/ whatever … of SEBit.

SEBit (www.sebit.world) is a start-up that cross over AI and VR.

Before I do startup, I worked in banks, have been doing Java development for FX trading for 10+ years.

So how did a Java developer start Python?


Once upon a time in PyCon 2016 ...
踏上這無盡旅途 ...

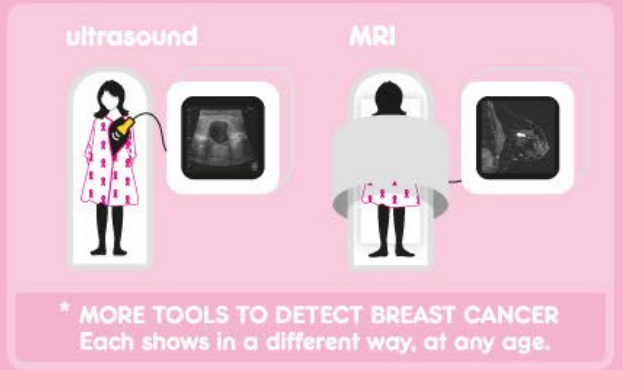
Learn Python through competitions

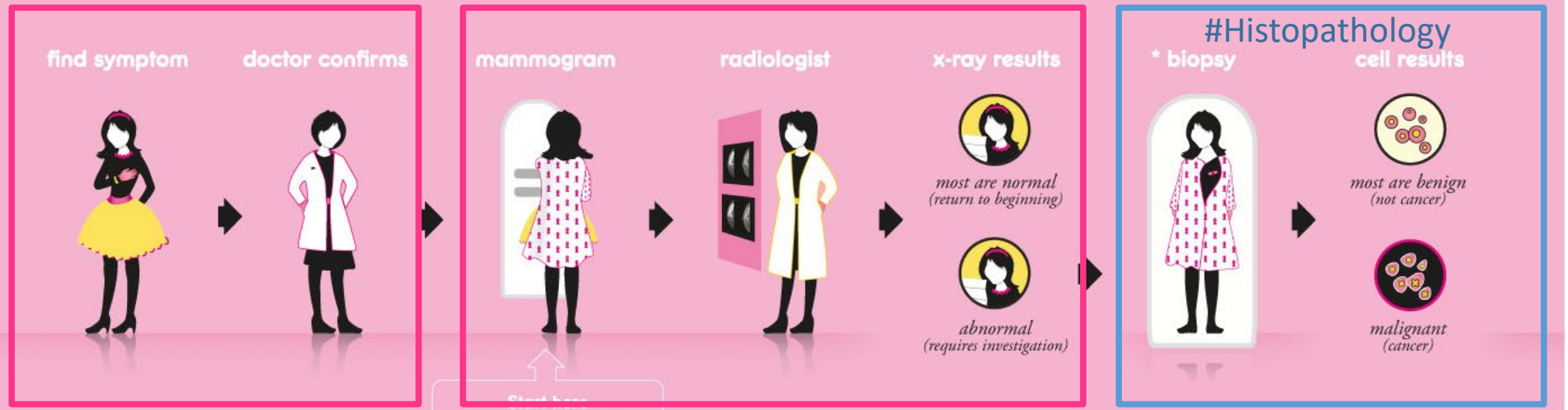
2015 - My first PyCon & started learning Python

Kaggle Competitions
- 2016 - Ultrasound Nerve Segmentation
- 2017 - The Nature Conservancy Fisheries Monitoring
- 2017 - Data Science Bowl Lung Cancer CT scan analysis
- 2017 - Intel & MobileODT Cervical Cancer Screening
- 2018 - iMaterialist Challenge Furniture at FGVC5

Biomedical Challenges
- 2017 - [LUNA16] - Lung Nodule CT scan analysis
- 2017 - [CAMELYON17] Classification of breast cancer metastases in lymph node
- 2018 - [BACH18] Grand Challenge on Breast Cancer Histology Images

Steps you can take to detect breast cancer





if you have no symptoms and are over age 40.

Depending on your risk and local guidelines, get screened every 1-3 years.
Decide how often with your doctor. Risk increases with age.
NHS invitation letter required in the UK.

"25-35% of breast cancers are found by patients. so reporting symptoms to a doctor is smart. However, mammograms can detect tiny lumps long before they can be felt. When found at earliest stages, the cure rate is up to 98%. Learn what breast cancer looks and feels like at worldwidebreastcancer.com."

Half the battle is knowing, the other half is going. Now you know, go.

worldwide breast cancer

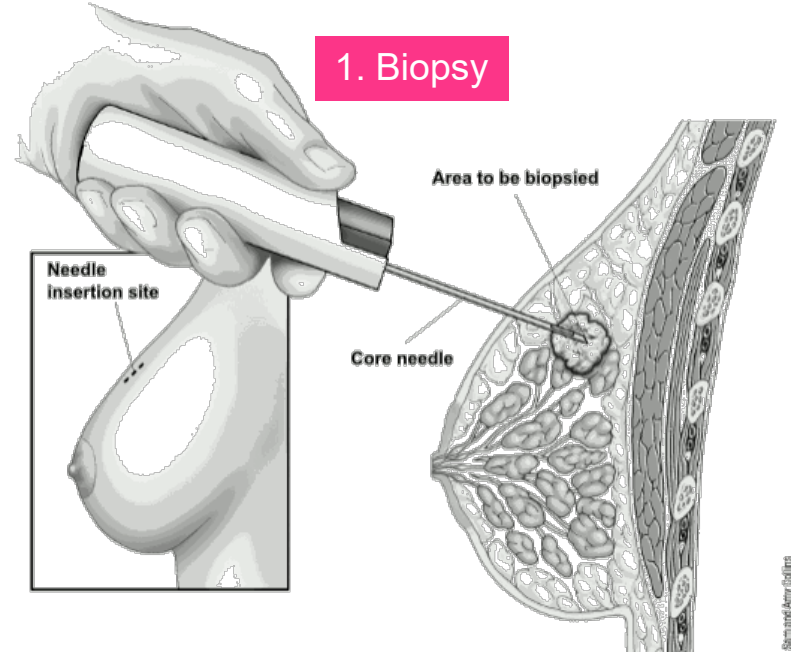
Did this poster speak to you? Learn more of what you can do about breast cancer screening through refreshing and informative design like this at http://worldwidebreastcancer.com

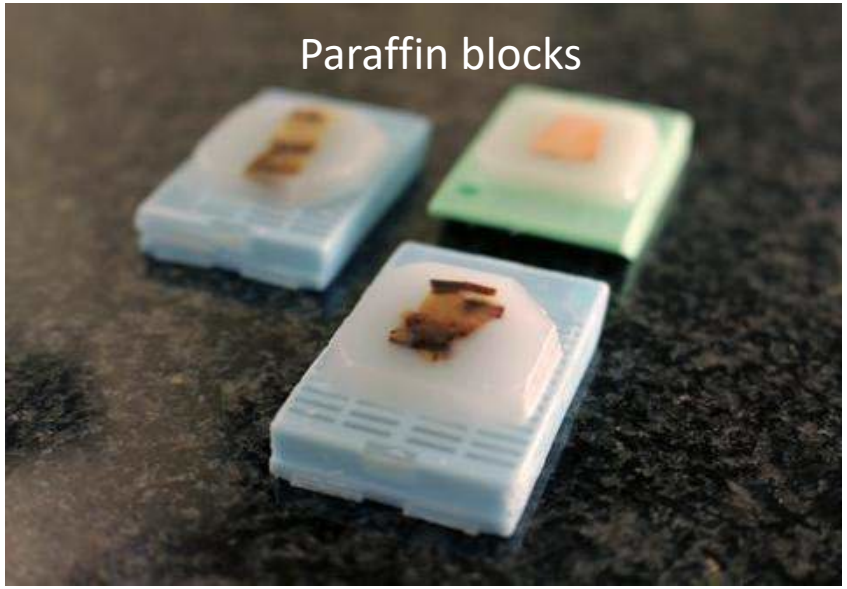
#Histopathology
組織病理學

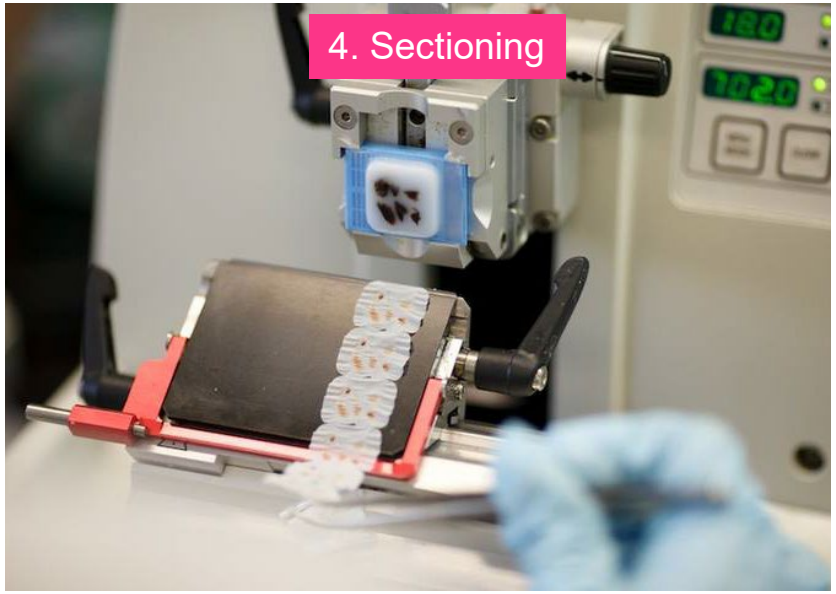
A branch of pathology, specialized in dealing with the histological structure of abnormal tissue

Image Courtesy : JMDHistology.com Histology slides

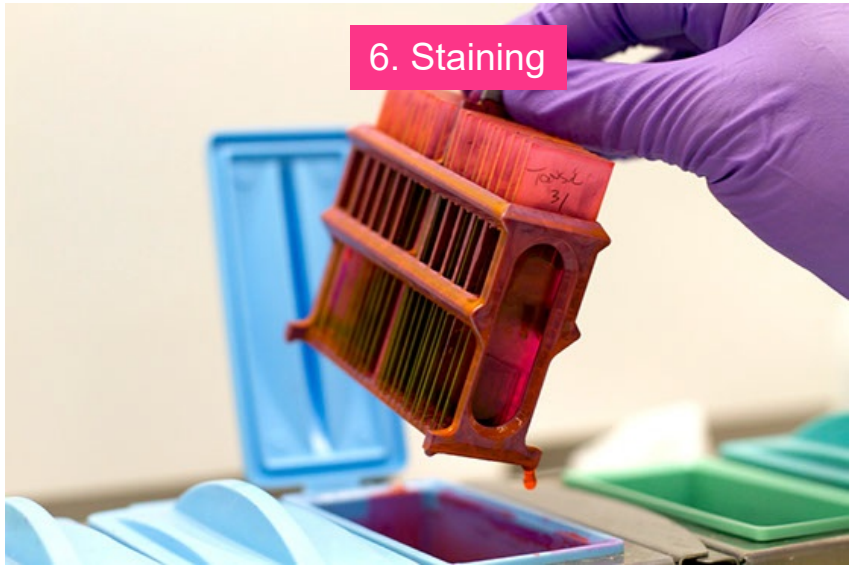
A biopsy is the only sure way to diagnose breast cancer.







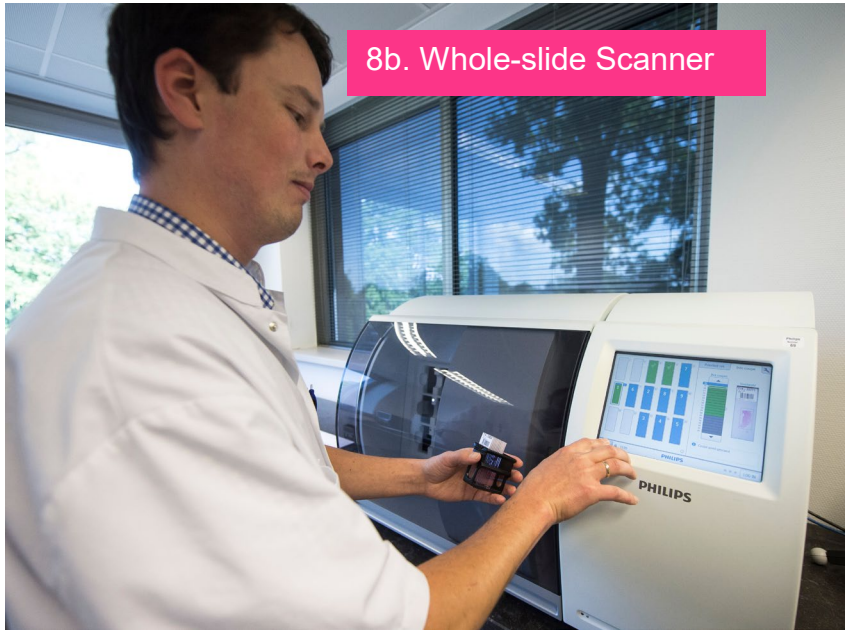
6. Staining



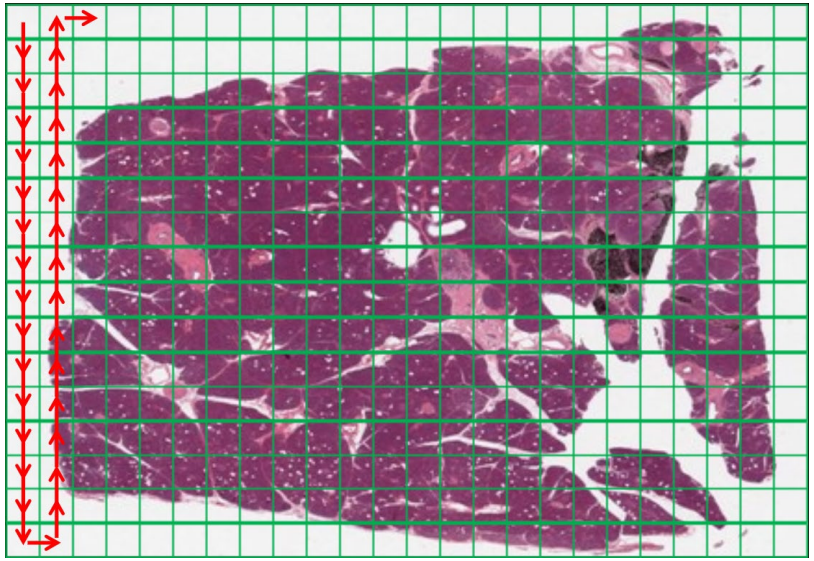
7. Glass slide ready

8a. Microscope

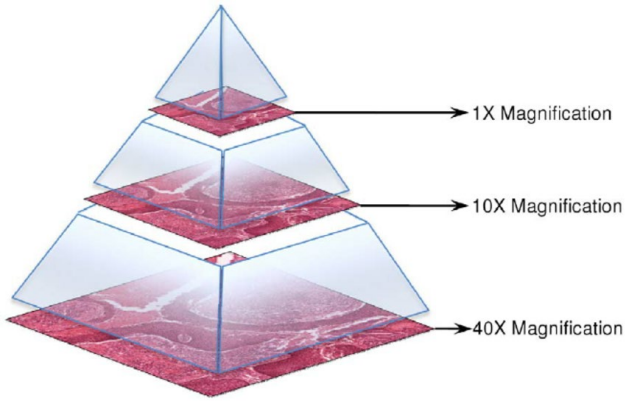
8b. Whole-slide Scanner



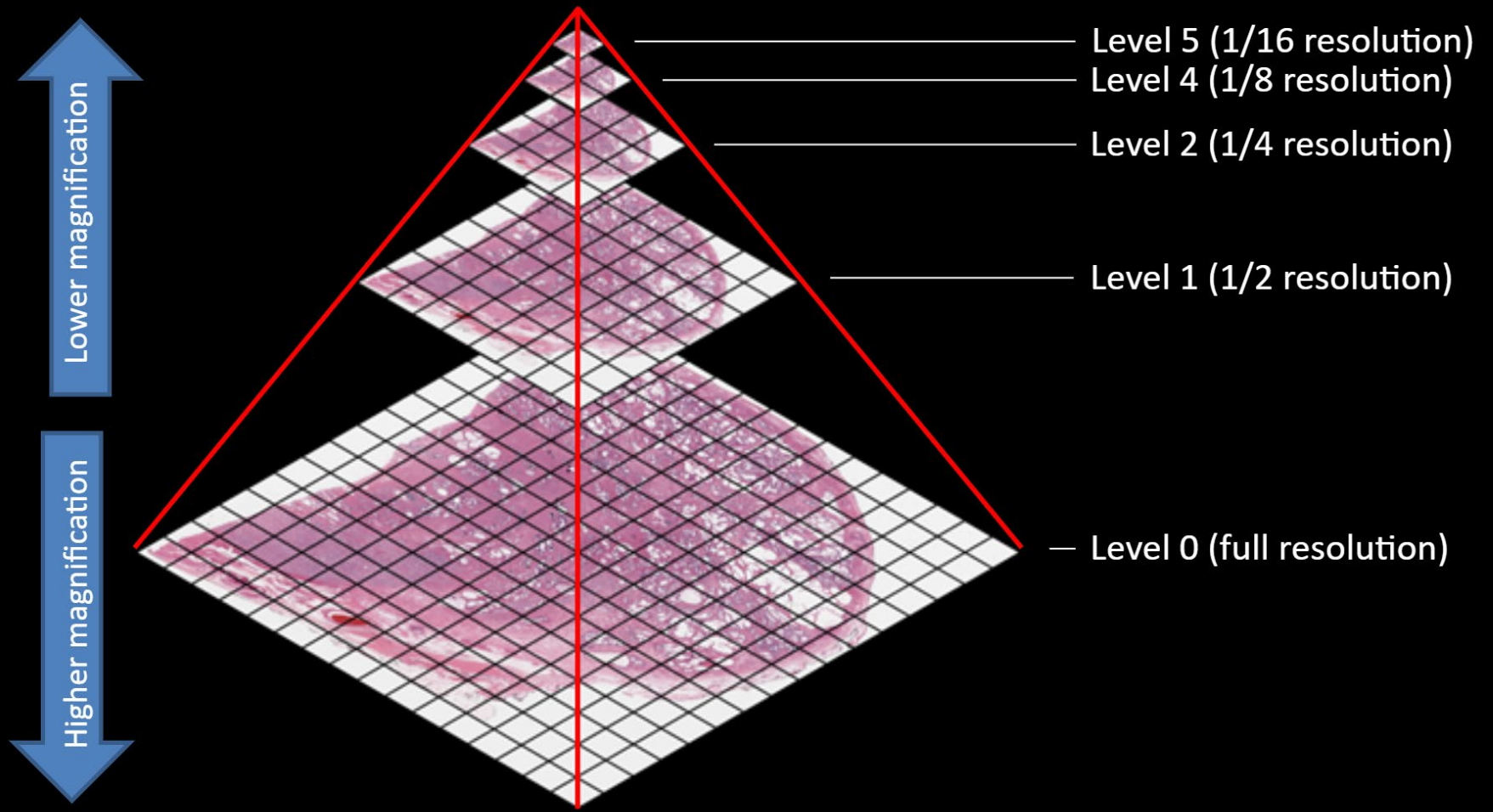
9. Scan



10. Whole-slide Image (WSI)



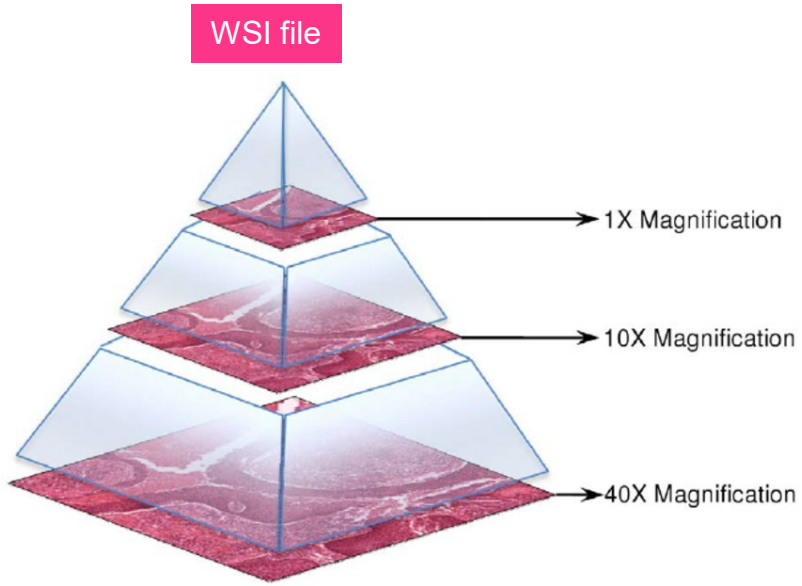
WSI file pyramid structure



Slide from: Selection and Implementation of a Digital Pathology System – by Toby C. Cornish, MD, PhD

Read WSI in Python

1. Unzip the binary to say: C:\openslide
2. pip install openslide-python
3. set PATH=C:\openslide\bin;%PATH%



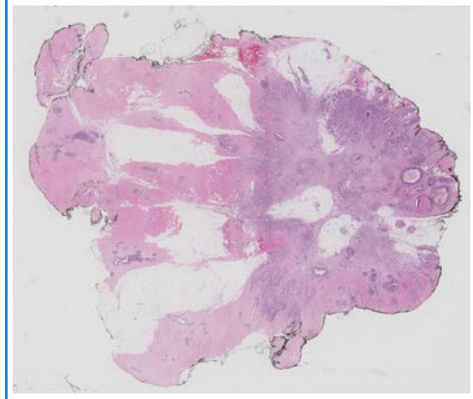
Jupyter QtConsole

File Edit View Kernel Window Help

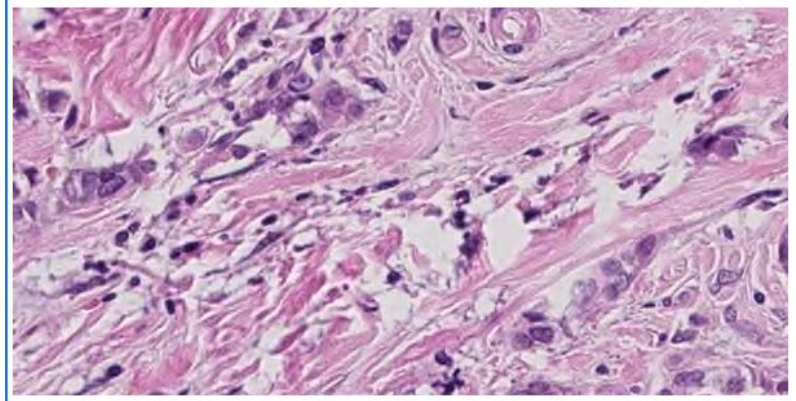
```
In [1]: from openslide import OpenSlide

In [2]: slide = OpenSlide('c:/tmp/A02.svs')

In [3]: slide.get_thumbnail((500,300))
Out[3]:
```



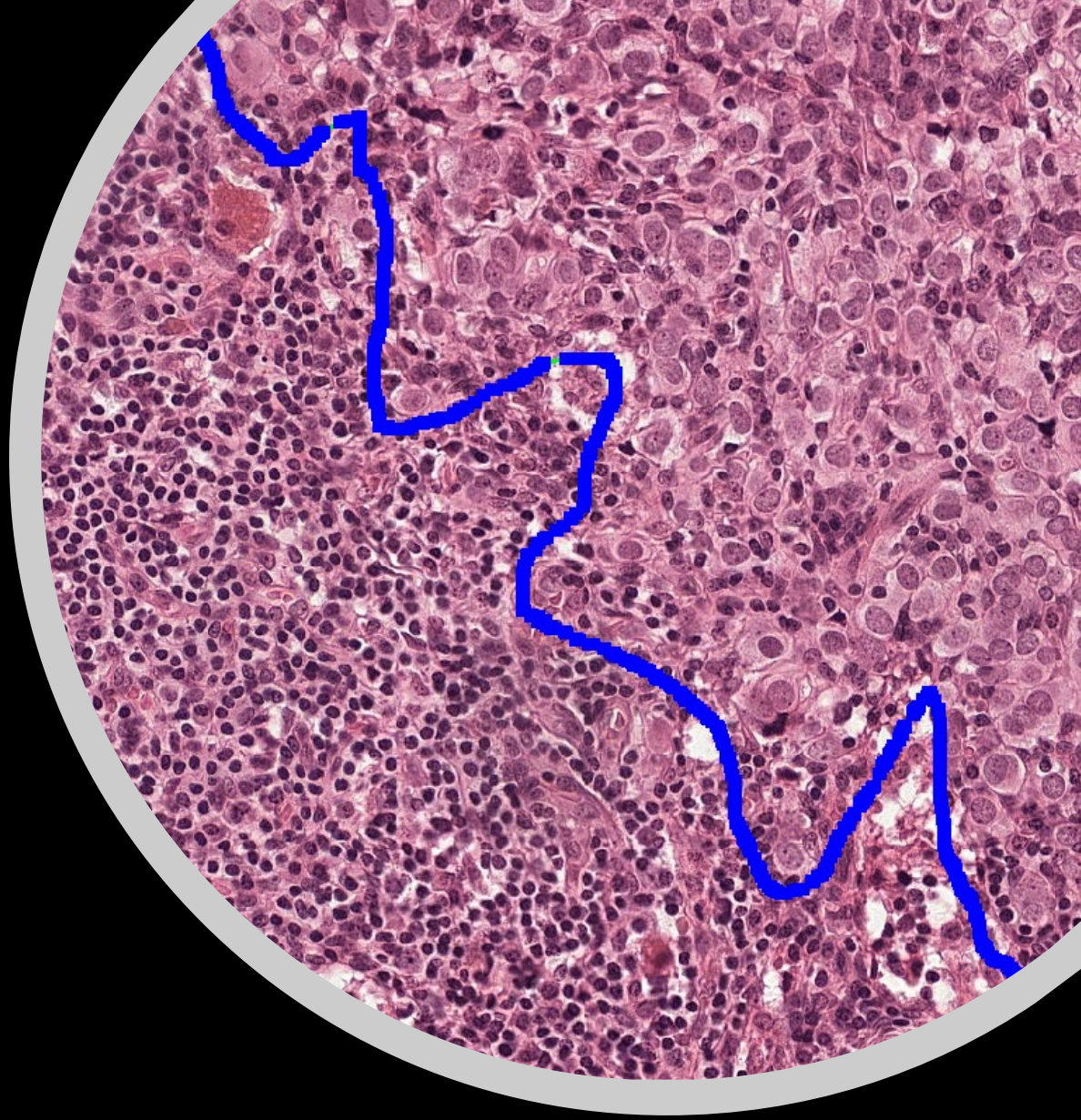
```
In [4]: slide.read_region(location=(27360,23152), level=0, size=(600,300))
Out[4]:
```



```
In [5]:
```

Difficulties in manual diagnosis

- It requires experienced pathologists
- Tedious and time consuming
- The result may varies depending on the expertise & judgement



ICIAR 2018 Grand Challenge on **Bre**A**st** **C**ancer **H**istology images

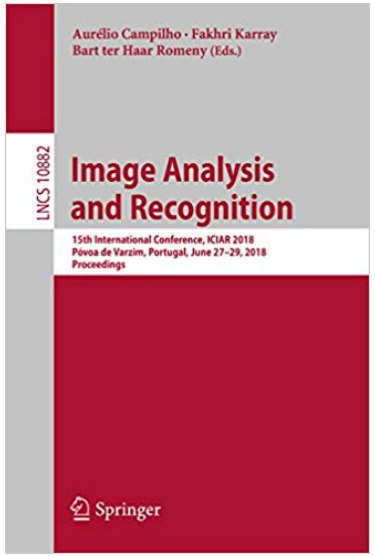
Objective

To develop methods for automatic detection of cancer regions in breast cancer histology images.

15th International Conference on Image Analysis and Recognition

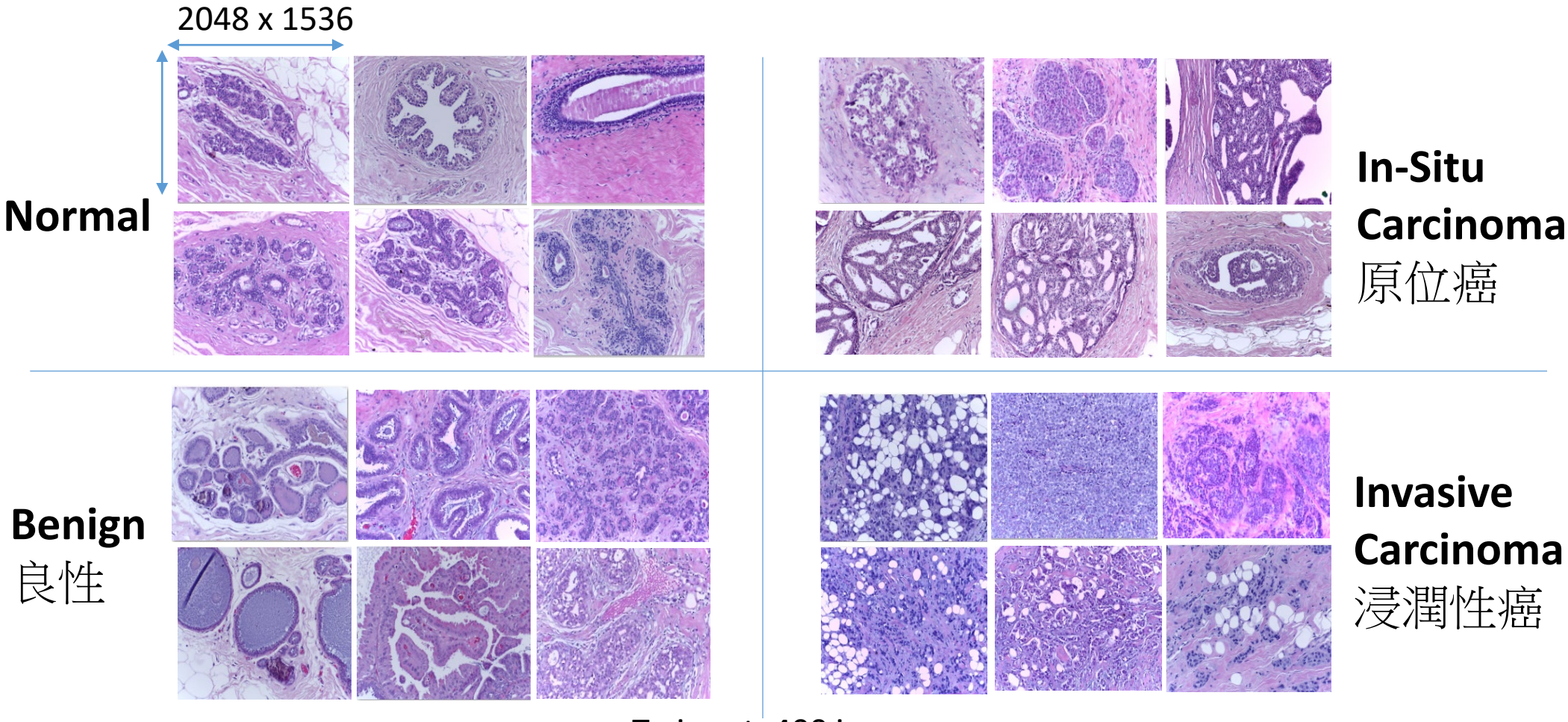
ICIAR 2018

June 27-29, 2018 – Póvoa de Varzim, Portugal



Dataset

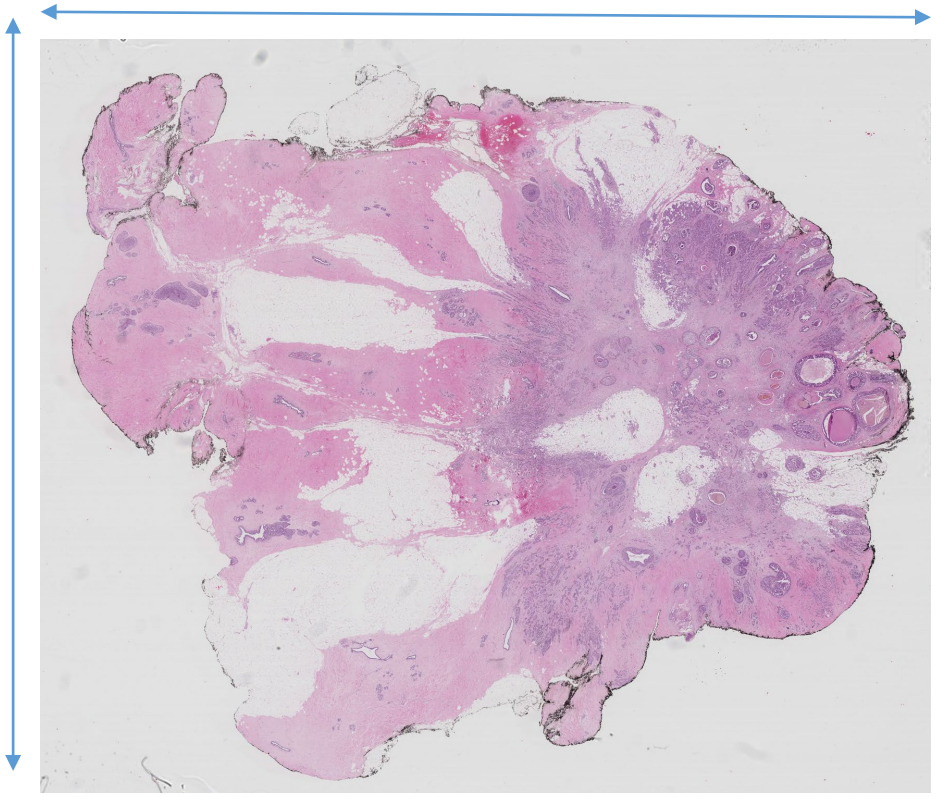
Part A: Four-classes classification of image patches



Train set: 400 images
Test set: 100 images

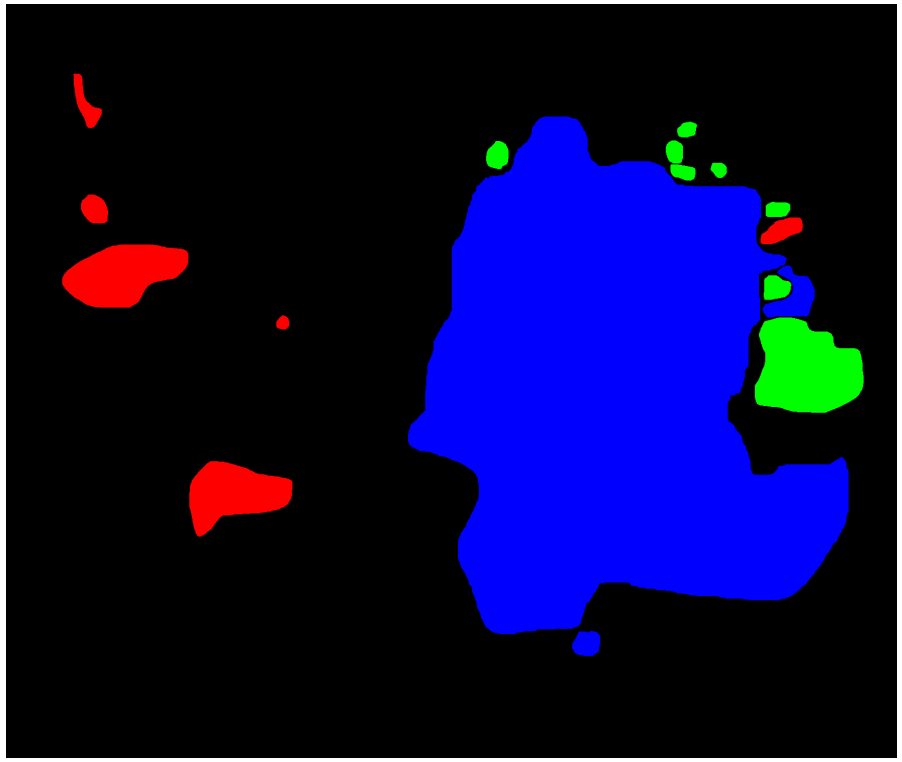
Part B: Apply the classification to WSI

~54000 x ~46000 pixels



Whole-slide image (WSI)

Train set: 10 WSI
Test set: 10 WSI



Ground truth

Results

Part A Leaderboard

Position = 1
Four-class Accuracy = 0.87

Position	Part A	First Author	Last Author	Country
1	0.87	Sai Saketh Chennamsetty	Varghese Alex	India
1	0.87	Scotty Kwok		Hong Kong SAR
3	0.86	Nadia Brancati	Daniel Riccio	Italy
4	0.84	Bahram Marami	Jack Zeineh	USA
5	0.83	Xianfei Zheng	Yang Duan	China
5	0.83	Matthias Kohl	Maximilian Baust	United Kingdom|Germany
5	0.83	Yaqi Wang	Jiannan Fang	China
8	0.81	J. Steinfeldt	S. Jabari	
8	0.81	Ismael Kone	Lahsen Boulmane	Morocco
8	0.81	Imane Nedjar	Mohammed Amine Chikh	Algeria|Belgium
11	0.8	Kamalakkannan Ravi	Mohanasankar Sivaprakasam	India
13	0.79	Zeya Wang	Eric P. Xing	USA
13	0.79	Hongliu CAO	Robert Sabourin	Canada|France
13	0.79	Kayoung Seo	Kyu-Hwan Jung	Korea
16	0.78	John-William Sidhom	Alexander S. Baras	USA
16	0.78	Yongxiang Huang		China
18	0.77	Yao Guo	Jun Liu	China
18	0.77	Nidhi Ranjan	A. D. Dileep	India
18	0.77	Amirreza Mahbod	Chunliang Wang	Austria|Sweden
21	0.76	Carlos A. Ferreira	Pedro Costa	Portugal
21	0.76	Gleb Makarchuk	Mikhail Belyaev	Russia
23	0.75	Mohammad Ibrahim Sarker	Dinar Akhmetzanov	South Korea|Russia
24	0.74	Alexander Rakhlin	Alexandr A. Kalinin	Russia|USA
25	0.72	Tomas Iesmantas	Robertas Alzbutas	Lithuania
25	0.72	Xinpeng Xie	Linlin Shen	China
		… total 51 entries		

Part B Leaderboard

Position = 1
Score = 0.6929

Position	Part B	First Author	Last Author	Country
1	0.6929	Scotty Kwok		Hong Kong SAR
2	0.5527	Bahram Marami	Jack Zeineh	USA
3	0.523	Yuanyuan Jia	Shaoqun Zeng	China
4	0.5203	Yaoqing Li	Guangzhe Dai	China
5	0.502	Masaki Murata	Masato Taki	Japan
6	0.501	Sameh Galal	Veronica Sanchez Freire	USA
7	0.4945	Quoc Dang Vu	Jin Tae Kwak	Korea
8	0.4681			
9	0.4637	Gleb Makarchuk	Mikhail Belyaev	Russia
10	0.4606	Xianfei Zheng	Yang Duan	China
11	0.4168	Matthias Kohl	Maximilian Baust	United Kingdom|Germany
12	0.3936	Alexey Ushakov	Kirill Emelyanov	Russia
13	0.3282	Vijay Narayanasamy	Devi Jeyachandran	USA

Computer vs Human

Reference:
Aresta et al, BACH: Grand Challenge on Breast Cancer Histology Images (https://arxiv.org/abs/1808.04277)

Table 4: Class-wise sensitivity and specificity of Part **A** approaches for the classes in study. Benchmarking results via fine-tuning are also shown. Acc - accuracy Se - sensitivity; Sp - specificity.

		Normal		**Benign**		***In situ***		**Invasive**	
Team	**Acc**	Se.	Sp.	Se.	Sp.	Se.	Sp.	Se.	Sp.
216 [20]	0.87	0.96	0.88	0.8	0.96	0.84	1.0	0.88	0.99
248 [21]	0.87	0.96	0.93	0.72	0.96	0.88	0.97	0.92	0.96
1 [22]	0.86	0.96	0.91	0.68	0.97	0.84	0.99	0.96	0.95
16 [23]	0.84	0.92	0.95	0.64	0.96	0.84	0.99	0.96	0.89
54 [25]	0.83	0.96	0.92	0.52	0.97	0.88	0.92	0.96	0.96
157 [26]	0.83	0.96	0.91	0.64	0.99	0.92	0.91	0.8	0.97
186	0.81	0.96	0.92	0.68	0.96	0.76	0.95	0.84	0.92
19 [27]	0.81	1.0	0.95	0.4	0.99	0.92	0.92	0.92	0.89
36	0.81	0.88	0.92	0.6	0.96	0.88	0.95	0.88	0.92
412	0.8	0.92	0.96	0.48	0.97	0.84	0.92	0.96	0.88
VGG	0.58	0.84	0.84	0.64	0.84	0.72	0.87	0.36	0.97
Inception	0.77	0.92	0.93	0.44	0.96	0.88	0.87	0.84	0.93
ResNet	0.76	0.88	0.92	0.52	0.95	0.8	0.87	0.84	0.95
DenseNet	0.79	0.92	0.96	0.36	0.99	0.92	0.83	0.96	0.95
Expert 1	0.96	0.96	0.99	0.92	0.97	1.0	1.0	0.96	0.99
Expert 2	0.94	0.96	0.99	0.88	0.96	1.0	1.0	0.92	0.97
Expert 3	0.78	0.88	0.99	0.76	0.79	0.56	0.97	0.92	0.96
Expert 4	0.73	0.40	0.99	0.84	0.71	0.76	0.97	0.92	0.97
Experts (avg.)	0.85± 0.10	0.80± 0.23	0.99± 0.00	0.85± 0.06	0.86± 0.11	0.83± 0.18	0.99± 0.01	0.93± 0.02	0.97± 0.01

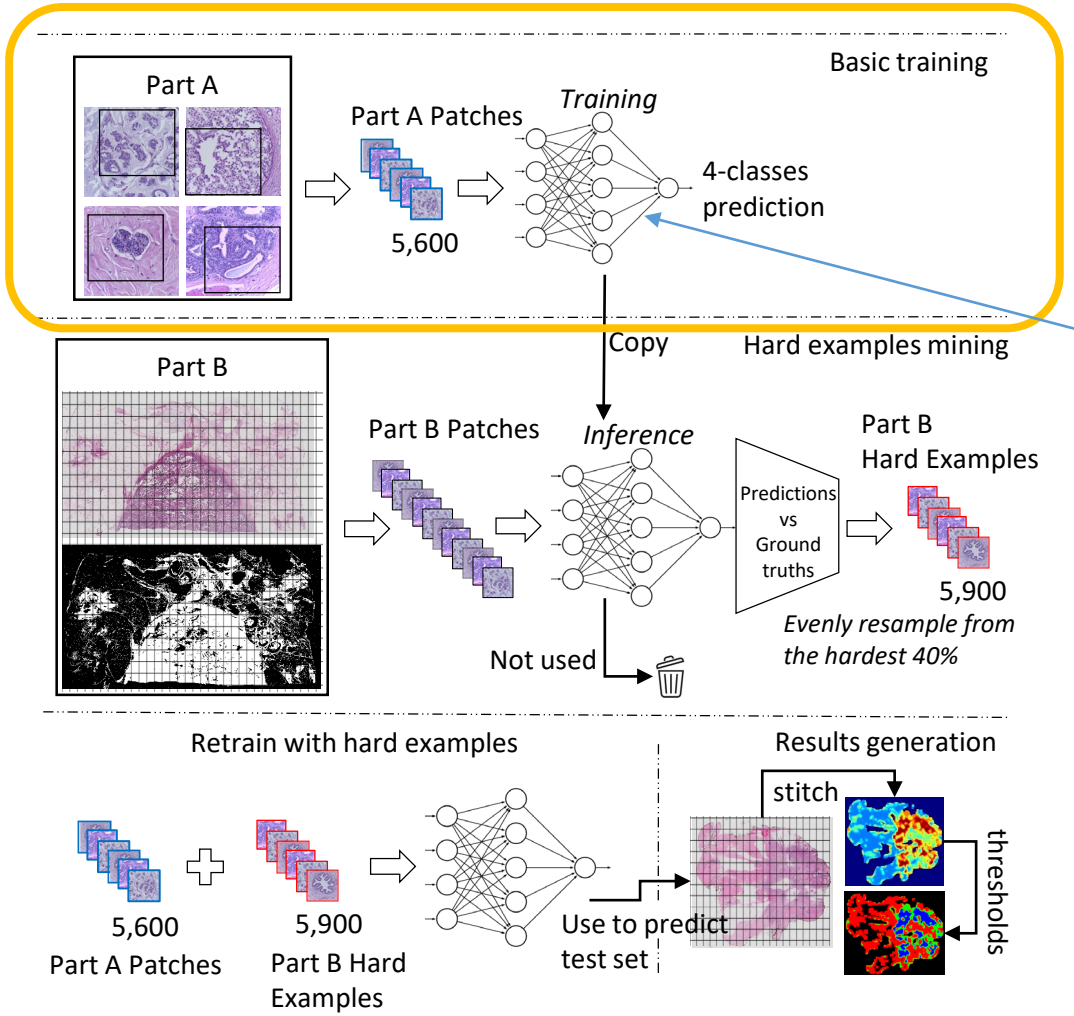
Models used by different teams

Reference:
Aresta et al, BACH: Grand Challenge on Breast Cancer Histology Images (https://arxiv.org/abs/1808.04277)

Team	Acc.	Approach
Chennamsetty *et al.* (216)[A] [20]	0.87	Resnet-101; Densenet-161
Kwok (248)[AB] [21]	0.87	Inception-Resnet-v2
Brancati *et al.* (1)[A] [22]	0.86	Resnet-34, 50, 101
Marami *et al.* (16)[AB] [23]	0.84	Inception-v3
Kohl *et al.* (54)[AB] [25]	0.83	Densenet-161
Wang *et al.* (157)[A] [26]	0.83	VGG16
Steinfeldt *et al.* (186)	0.81	XCeption
Kone *et al.* (19)[A] [27]	0.81	ResNeXt50
Nedjar *et al.* (36)	0.81	Inception-v3, Resnet-50, MobileNet
Ravi *et al.* (412)	0.8	Resnet-152
Wang *et al.* (22) [29]	0.79	VGG16
Cao *et al.* (425) [31]	0.79	PTFAS+GLCM, ResNet-18, ResNeXt, NASNet-A, ResNet-152, VGG16, Random Forest SVM
Seo *et al.* (60)	0.79	ResNet, Inception-V3, Random Forests
Sidhom *et al.* (370)	0.78	ResNet-50
Guo *et al.* (242) [33]	0.77	GoogLeNet
Ranjan *et al.* (61)	0.77	AlexNet
Mahbod *et al.* (73) [34]	0.77	ResNet-50, ResNet-101
Ferreira *et al.* (18) [35]	0.76	Inception-ResNet-v2
Pimkin *et al.* (256) [36]	0.76	ResNet34, Densenet169, Densenet201 XGBoost
Sarker *et al.* (358)	0.75	Inception-v4
Rakhlin *et al.* (98) [37]	0.74	VGG16, ResNet-50, IcenptionV3, LightGBM

Methodology

1. Train a basic classifier using part A

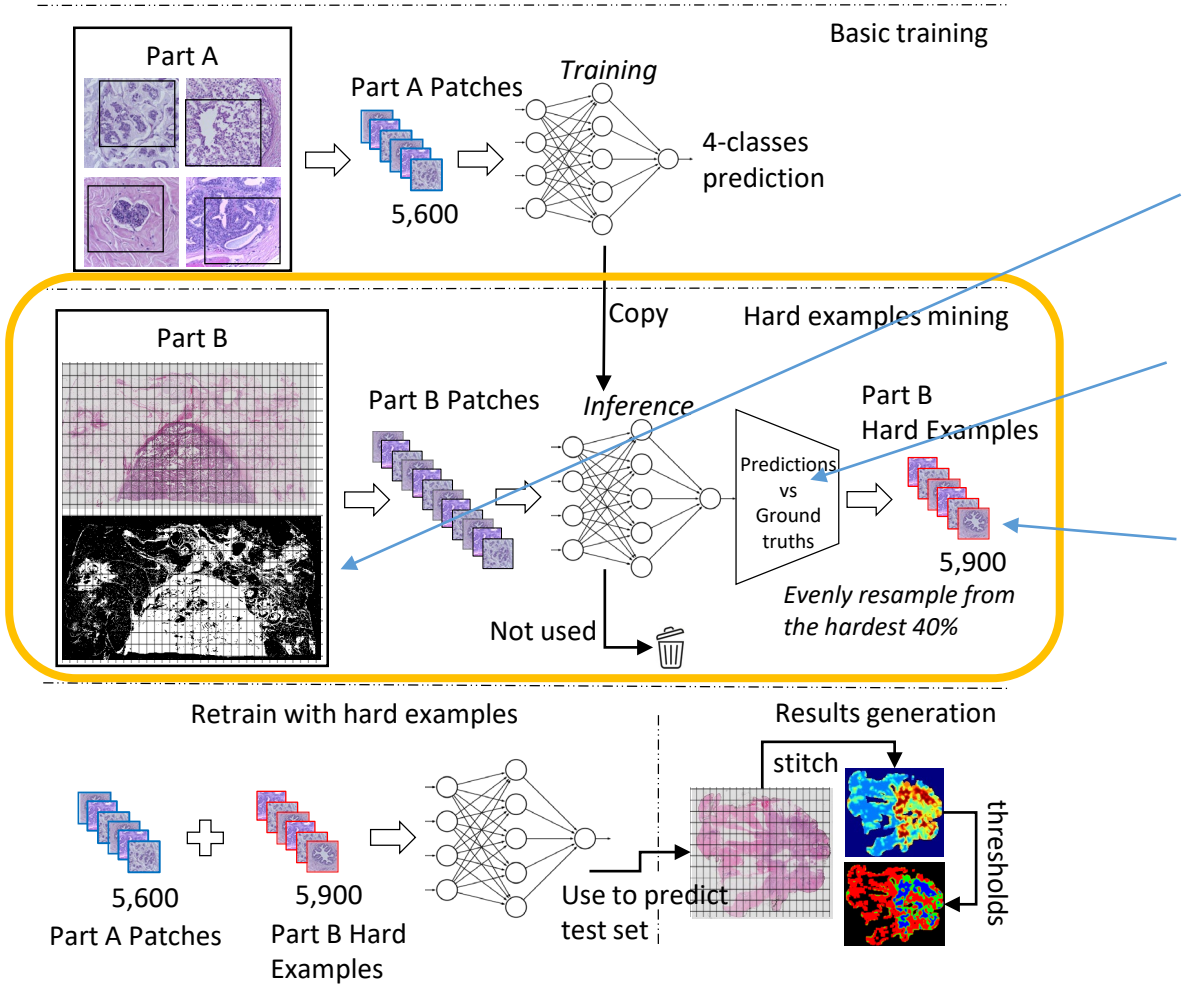


Accuracy of different models

Model	Four-classes Accuracy	Binary Accuracy
VGG19	0.70	0.81
Inception-v3	0.74	0.85
Inception-v4	0.71	0.82
Inception-Resnet-v2	0.79	0.91

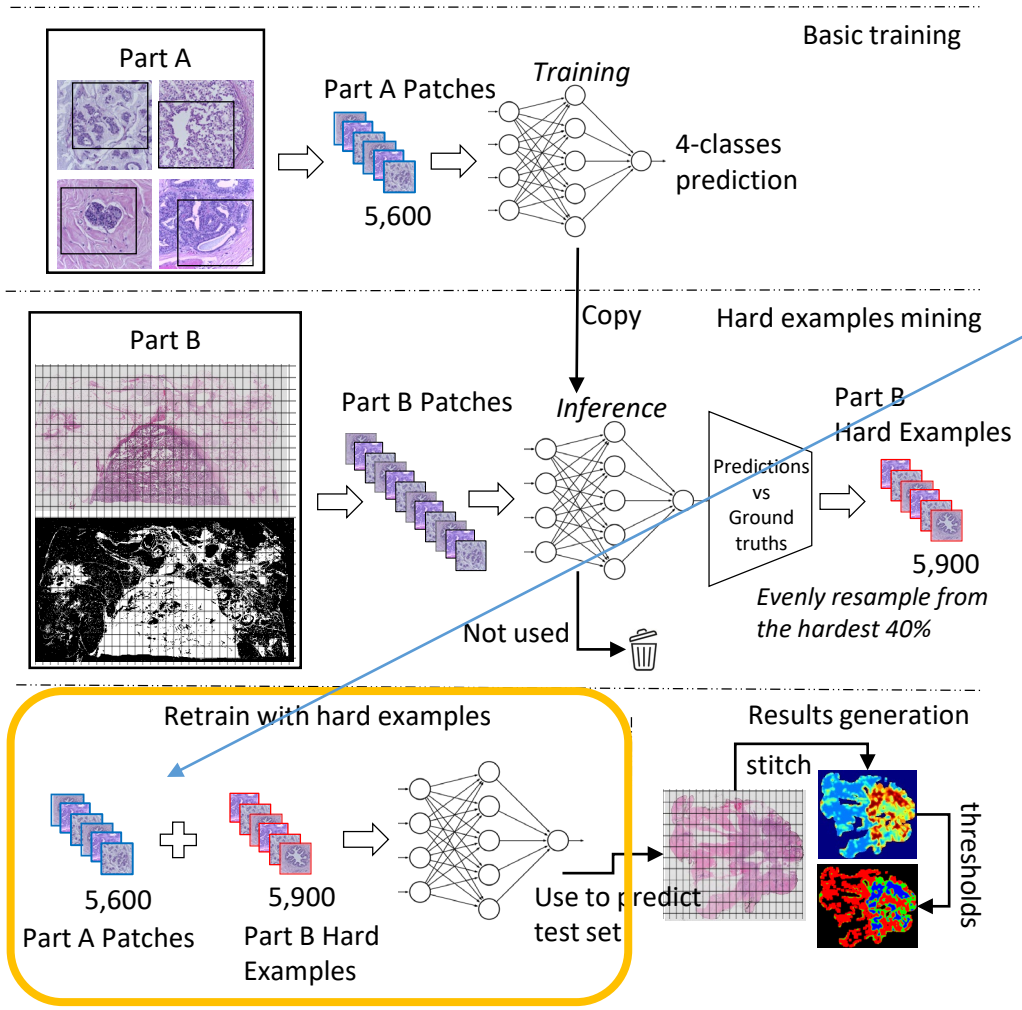
- Sliding window patch extraction
- Augmentations: 90 deg rotation, flipping, HSV color perturbation
- Fine-tuning using pre-trained ImageNet.
- Use 3-folds CV
- Using 2 GPUs (GTX 1080 Ti), the model converged within 25 epochs, in <35 mins.
- Inception-Resnet-v2, CV accuracy 0.79

2. Extract hard examples from part B



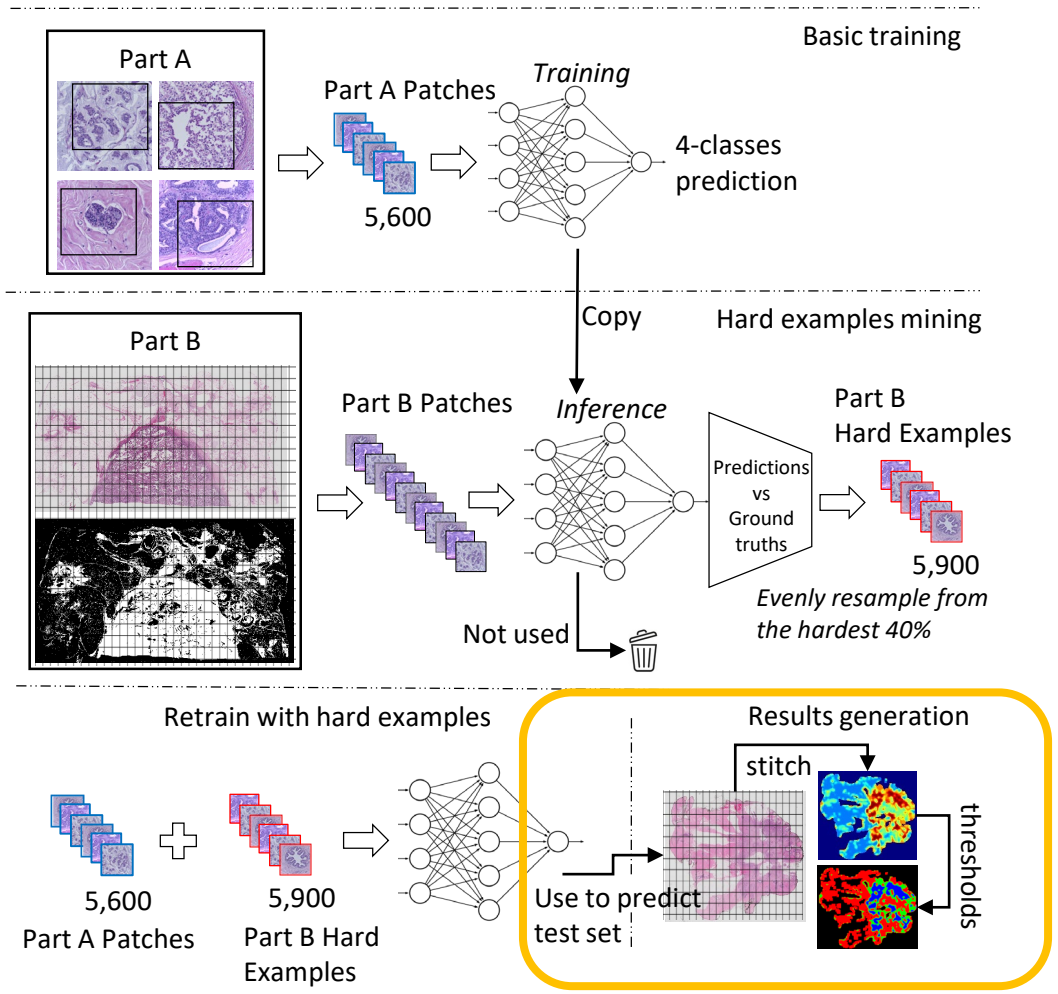
- Foreground extraction in LAB color space. And sliding window patch extraction
- Use prediction error to rank the difficulties of each patch
- A total of 5,900 patches were evenly sampled from each classes of hard examples

3. Combine A+B to train the final classifier

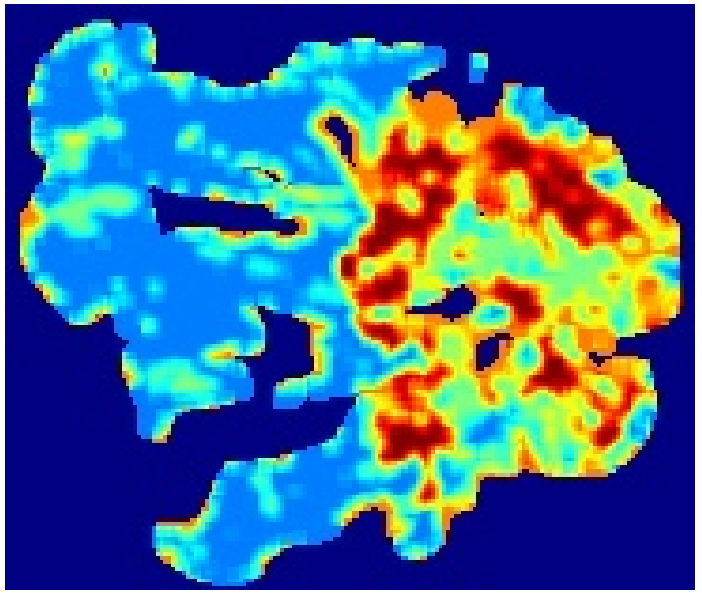


- Combine part A patches (5,600) and part B hard examples (5,900)
- Use the same model and hyper parameters
- Model converged within 15 epochs, in <40 mins.

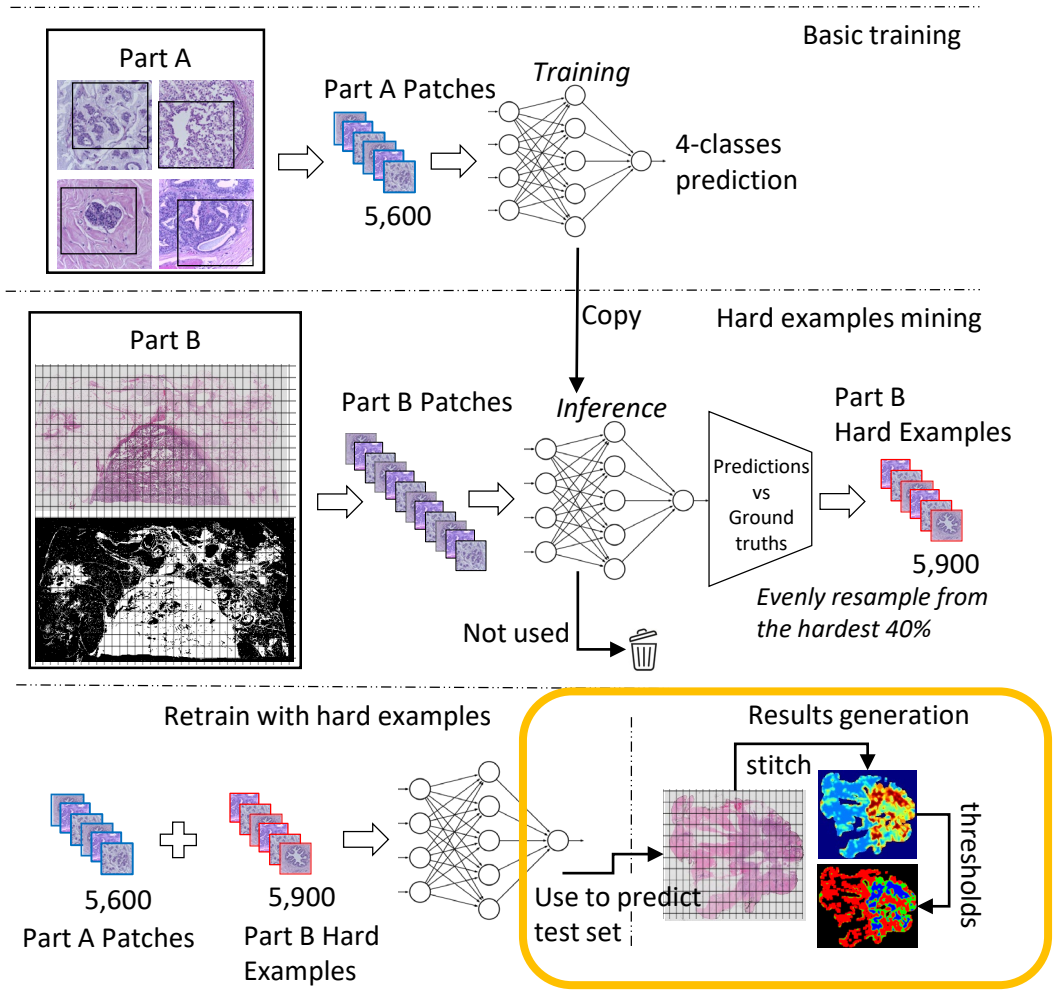
4. Generate heatmap



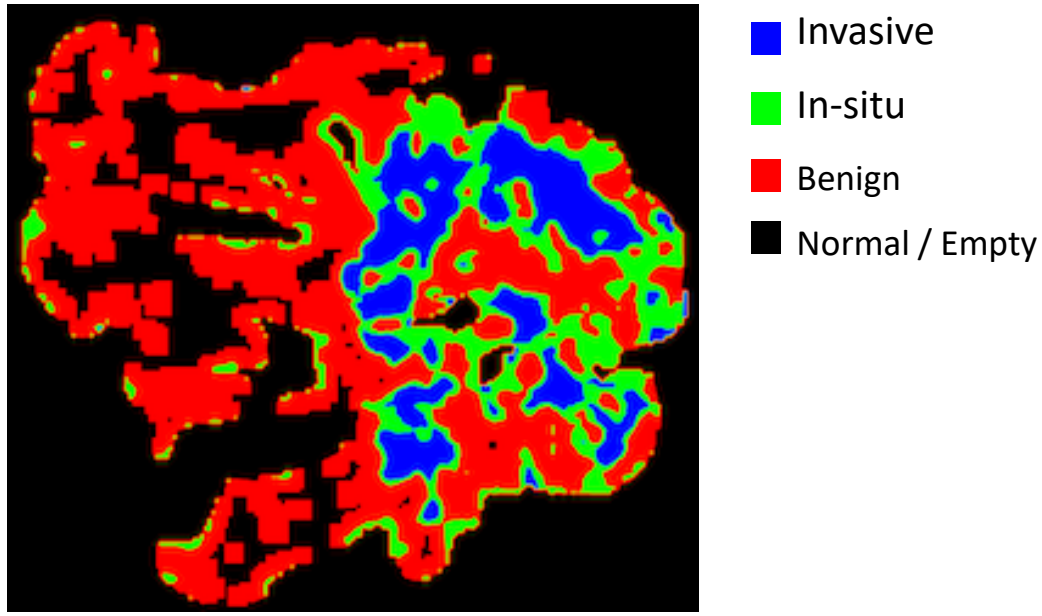
- Use final classifier to infer patch-wise predictions
- Stitch the patch-wise prediction onto a heatmap
- Each class map to an intensity value: Normal=0 , Benign=1, In-situ=2 and Invasive=3



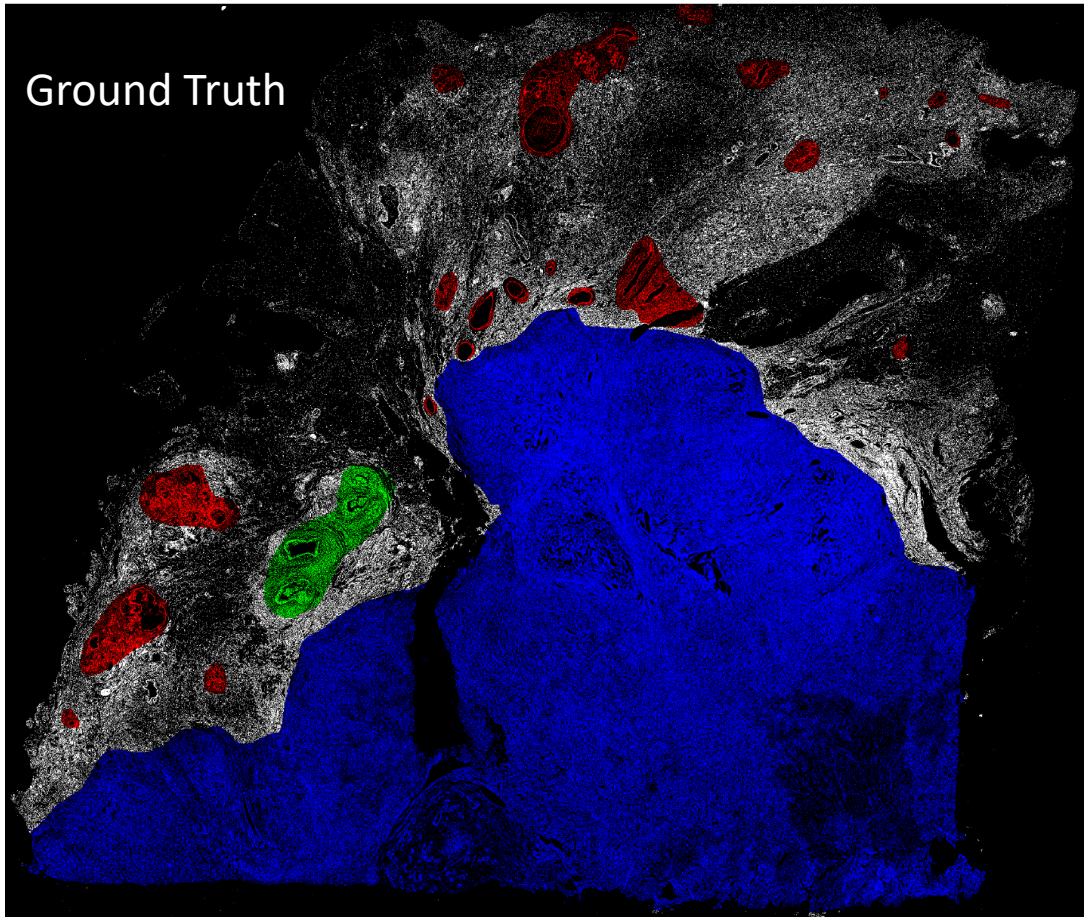
5. Apply thresholds

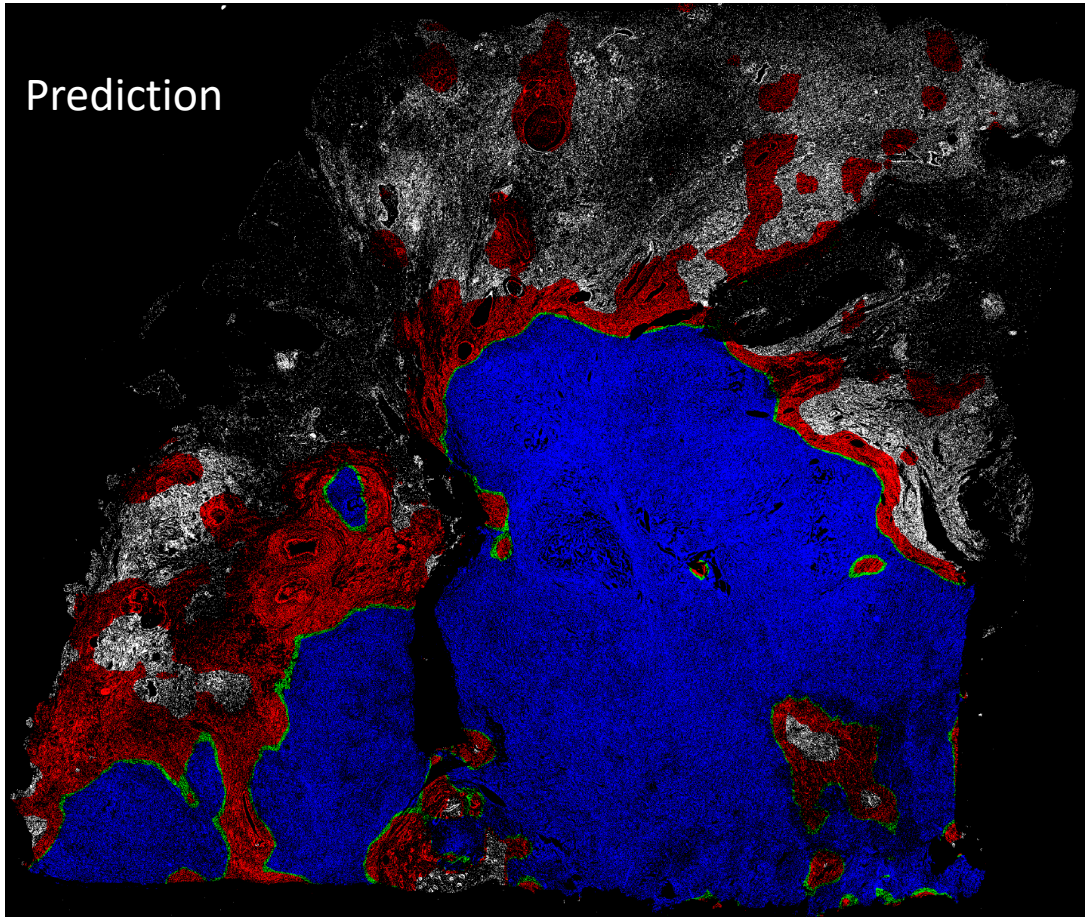


- Apply thresholds to the heatmap to give the class map



Part B sample results





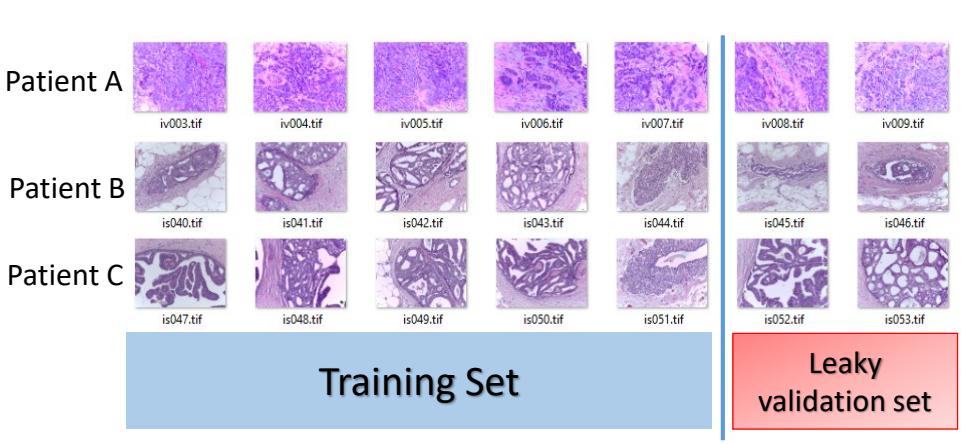
Cross Validation

Lesson learnt #1

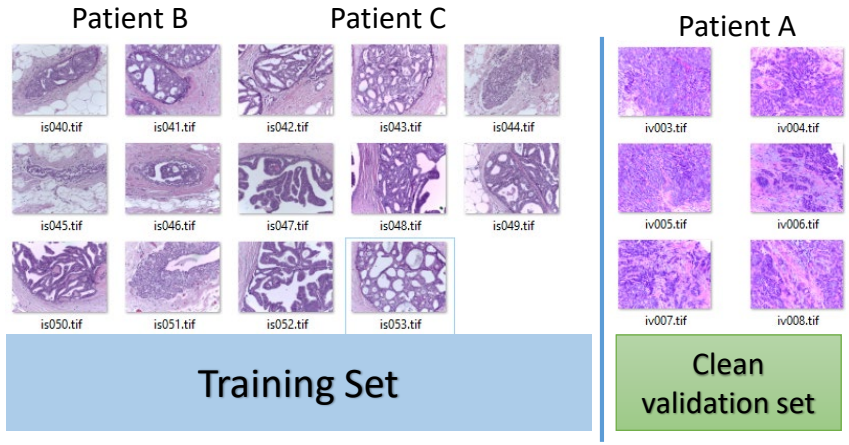
Cross validation could be very tricky

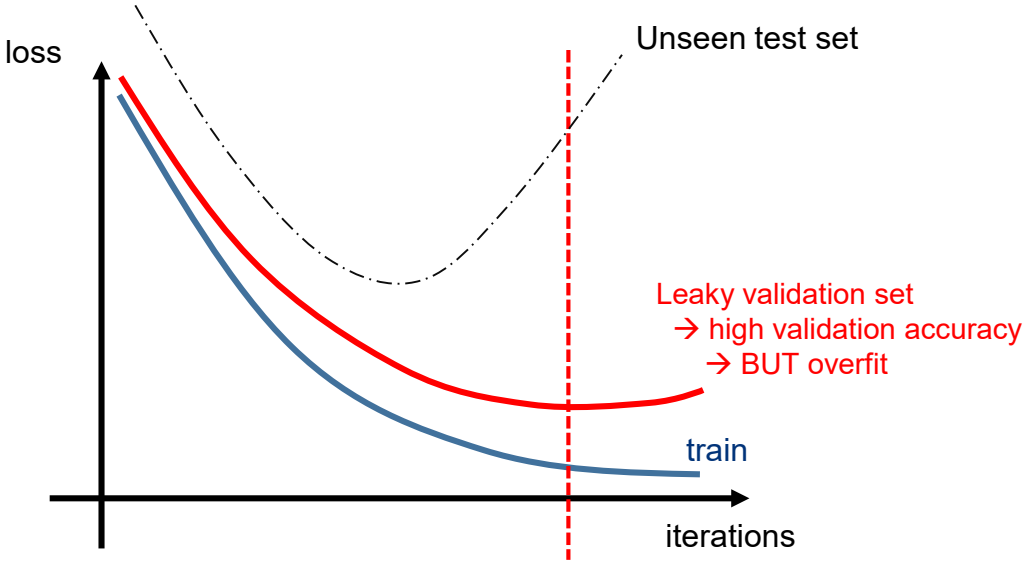
Part A Position	CV (Validation accuracy)	Leaderboard (Actual accuracy)	Under/Over estimate
1	0.97	0.87	-10%
1	0.79	0.87	8%
3	0.97	0.86	-11%
5	0.94	0.83	-11%
13	0.87	0.79	-8%
18	0.89	0.77	-12%
21	0.90	0.76	-14%
24	0.87	0.74	-13%
25	0.87	0.72	-15%
31	0.98	0.66	-32%
37	0.95	0.61	-34%

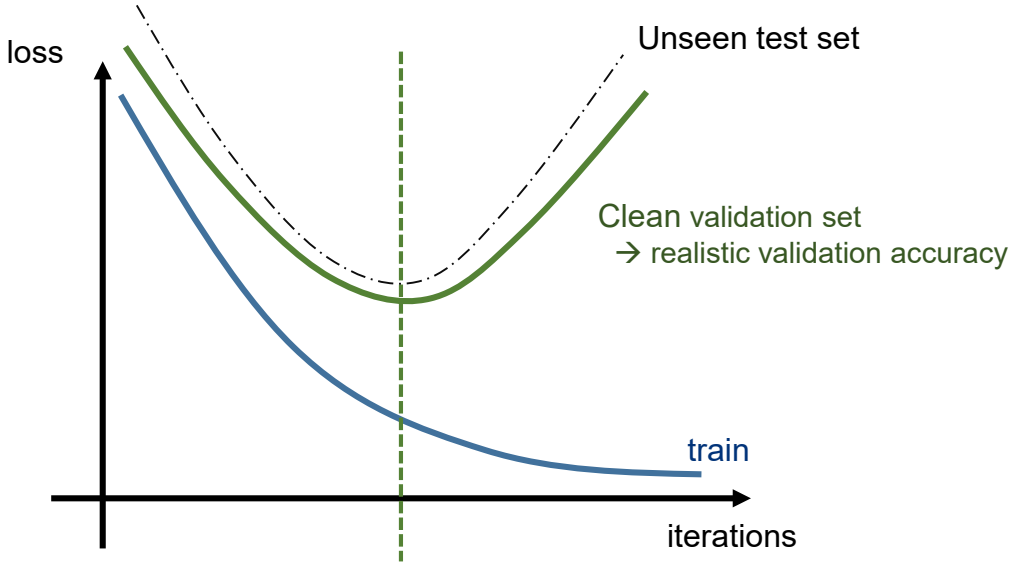
A “leaky” validation split induce overfitting



vs





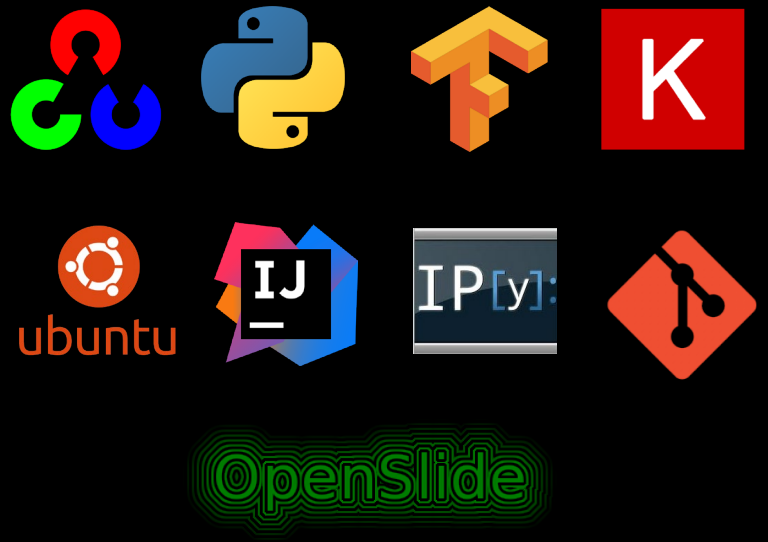


Software

Lesson learnt #2

Software stack

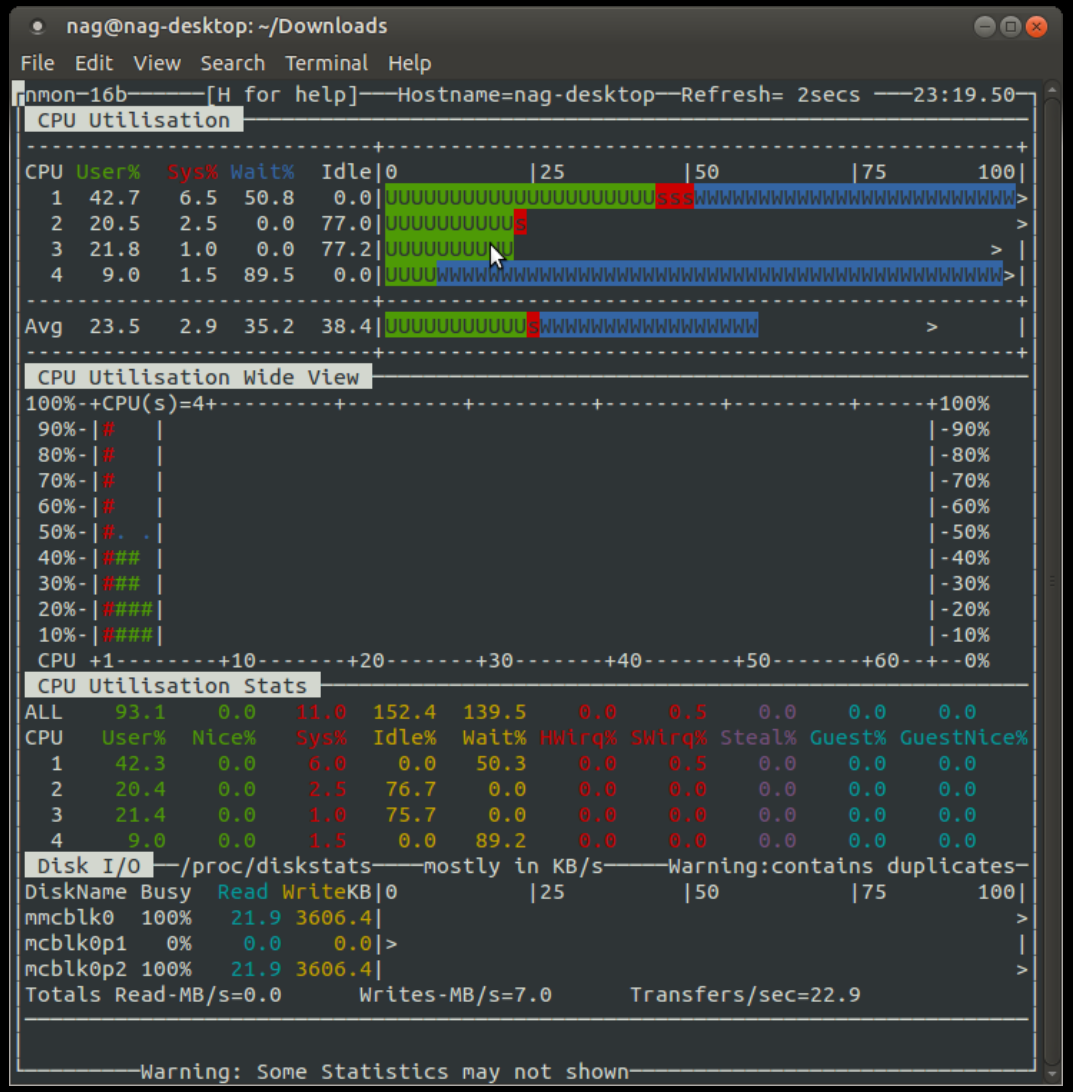
- Python
  - Keras (Tensorflow backend)
  - OpenCV
  - Openslide Python
  - Pandas, Numpy, PIL, etc.
- IntelliJ, Git
- Qt Console
- Ubuntu



Software stack … opinions

- Pandas, Numpy, OpenCV → inevitable
- IntelliJ → shortcuts, refactoring, debugger → higher productivity
- Git → code diff, merge, history → ease of mind
- Linux vs Windows vs Anaconda → It depends. If you need the latest version of OpenCV + Pytorch + Keras + Tensorflow + CUDA + cudnn, then Linux
- Keras → Data Iterator , e.g. DirectoryIterator, NumpyArrayIterator, etc → learn it & may be create your own

nmon + nvidia-smi




```
+-----------------------------------------------------------------------------+
| NVIDIA-SMI 378.13                 Driver Version: 378.13                    |
|-------------------------------+----------------------+----------------------+
| GPU  Name        Persistence-M| Bus-Id        Disp.A | Volatile Uncorr. ECC |
| Fan  Temp  Perf  Pwr:Usage/Cap|         Memory-Usage | GPU-Util  Compute M. |
|===============================+======================+======================|
|   0  GeForce GTX 1080    Off  | 0000:01:00.0      On |                  N/A |
| 34%   52C    P2    39W / 180W |   2553MiB /  8112MiB |     14%      Default |
|-------------------------------+----------------------+----------------------+
|   1  GeForce GTX 1080    Off  | 0000:02:00.0     Off |                  N/A |
| 27%   36C    P8     7W / 180W |      1MiB /  8114MiB |      0%      Default |
+-------------------------------+----------------------+----------------------+

+-----------------------------------------------------------------------------+
| Processes:                                                       GPU Memory |
|  GPU       PID  Type  Process name                               Usage      |
|=============================================================================|
|    0      1292    G   /usr/lib/xorg/Xorg                          1719MiB |
|    0      1760    G   compiz                                       386MiB |
|    0      2431    G   ...nabled/SubresourceFilter/EnabledForPhishi   446MiB |
+-----------------------------------------------------------------------------+
```

Keras – Multi GPU Support

- Quasi-linear speedup on up to 8 GPUs.

```
from keras.utils import multi_gpu_model

# Replicates `model` on 8 GPUs.
# This assumes that your machine has 8 available GPUs.
parallel_model = multi_gpu_model(model, gpus=8)
parallel_model.compile(loss='categorical_crossentropy',
                       optimizer='rmsprop')

# This `fit` call will be distributed on 8 GPUs.
# Since the batch size is 256, each GPU will process 32 samples.
parallel_model.fit(x, y, epochs=20, batch_size=256)
```

Keras – Save/Load a parallel model

```python
from keras.applications.inception_resnet_v2 import InceptionResNetV2
from keras.utils import multi_gpu_model

# Original model
base_model = InceptionResNetV2(...)

# Parallel model
parallel_model = multi_gpu_model(base_model, gpus=4)

# Save
checkpoint = MultiGPUCheckpointCallback(base_model, ...)

# Load
base_model.load_weights('/tmp/weights.h5')

# Training
parallel_model.compile(...)

parallel_model.fit_generator(...)
```

Keras – MultiGPU CheckpointCallback

```python
class MultiGPUCheckpointCallback(Callback):

    def __init__(self, filepath, base_model, monitor='val_loss', verbose=0,
                 save_best_only=False, save_weights_only=False,
                 mode='auto', period=1):
        super(MultiGPU_Checkpoint_Callback, self).__init__()
        self.base_model = base_model
        self.monitor = monitor
        self.verbose = verbose
        self.filepath = filepath
        self.save_best_only = save_best_only
        self.save_weights_only = save_weights_only
        self.period = period
        self.epochs_since_last_save = 0

        if mode not in ['auto', 'min', 'max']:
            warnings.warn('ModelCheckpoint mode %s is unknown, '
                          'fallback to auto mode.' % (mode),
                          RuntimeWarning)
            mode = 'auto'

        if mode == 'min':
            self.monitor_op = np.less
            self.best = np.Inf
        elif mode == 'max':
            self.monitor_op = np.greater
            self.best = -np.Inf
        else:
            if 'acc' in self.monitor or self.monitor.startswith('fmeasure'):
                self.monitor_op = np.greater
                self.best = -np.Inf
            else:
                self.monitor_op = np.less
                self.best = np.Inf

    def on_epoch_end(self, epoch, logs=None):
        logs = logs or {}
        self.epochs_since_last_save += 1
```

Source:
https://github.com/keras-team/keras/issues/8463#issuecomment-345914612

Hardware

Lesson learnt #3

My deep learning hardware journey

"The AI Expert" build

……

Indie build

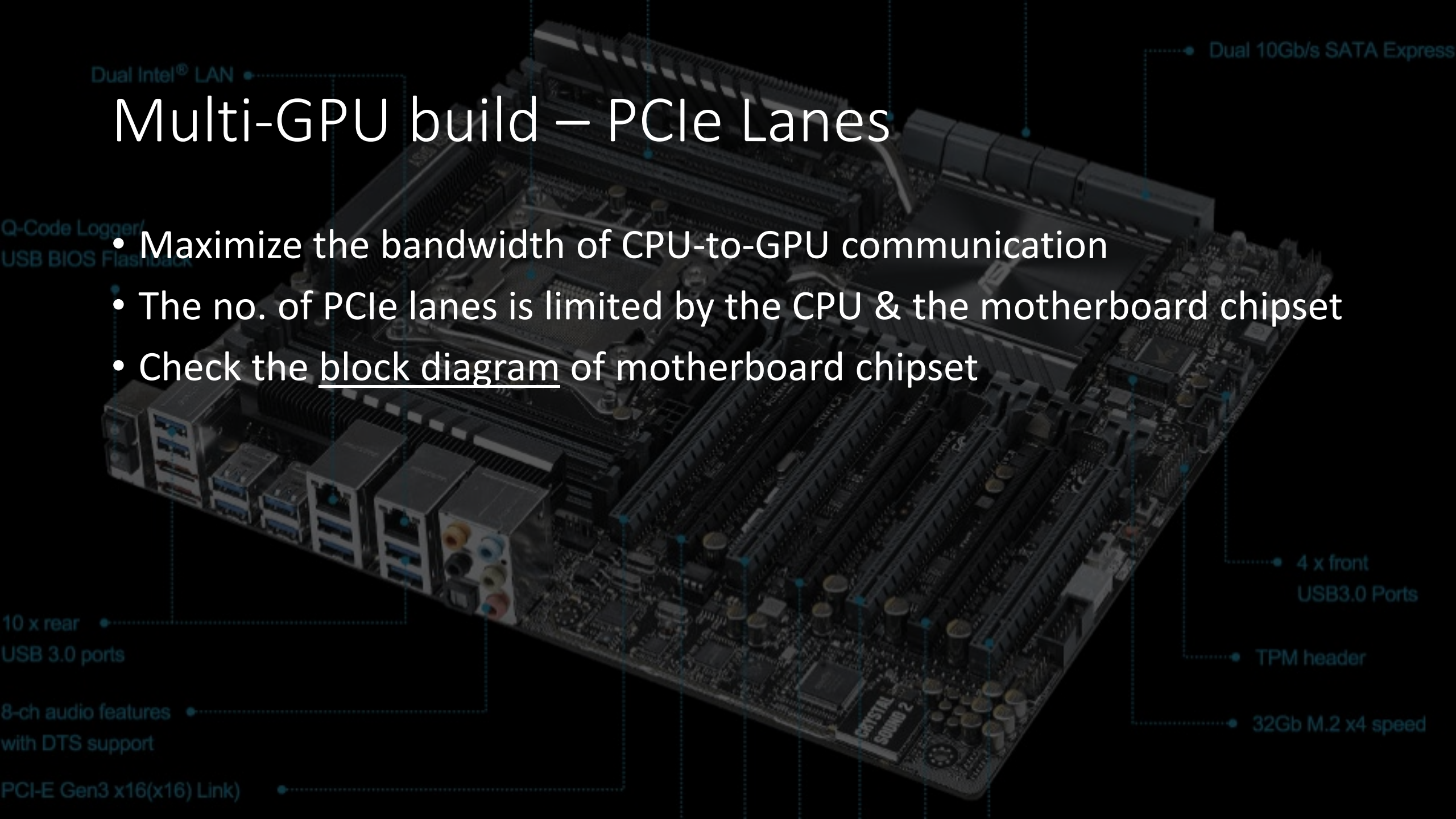
Mini ITX
650W PSU
GTX 1070 (8 GB)
$15K HKD

Hobbyist build

Full tower
1600W PSU
GTX 1080Ti (11GB) (x1 - x4)
$25K – $45K HKD

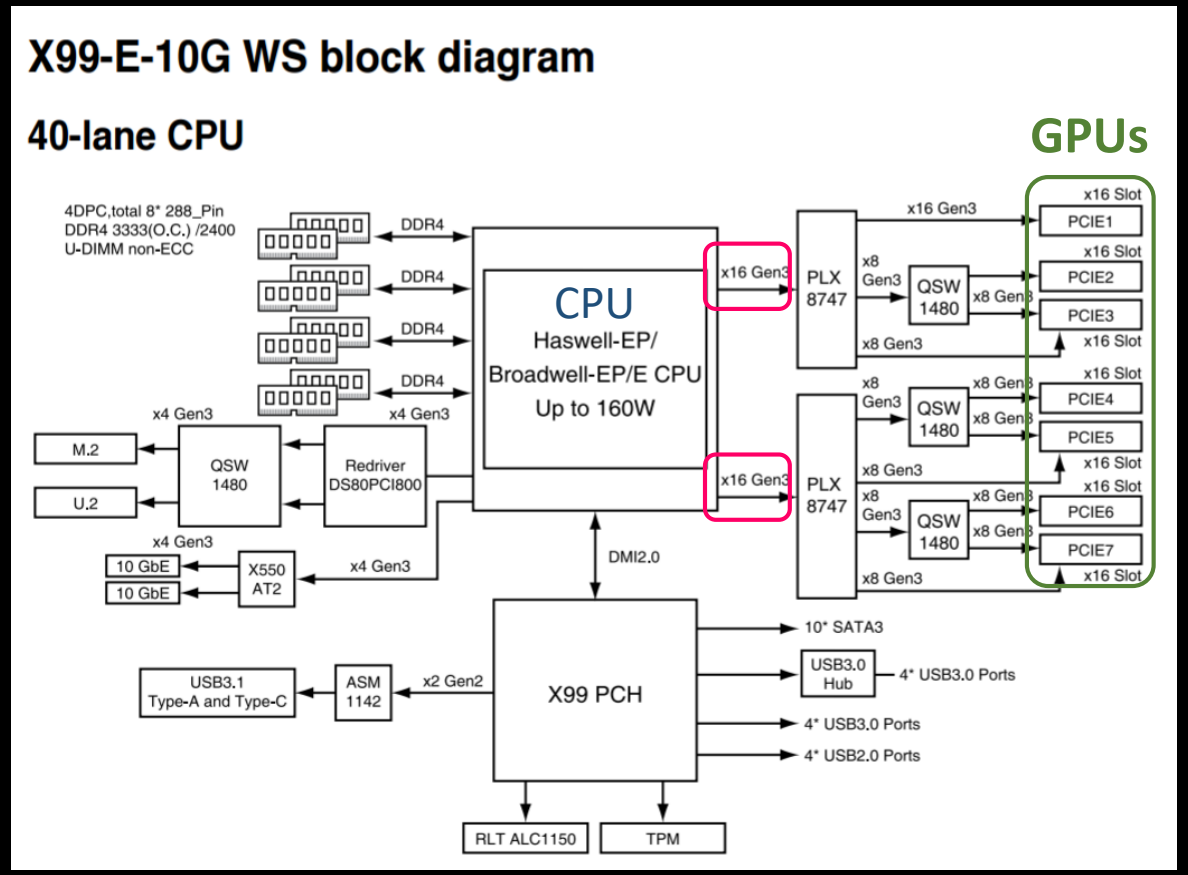
Multi-GPU build – PCIe Lanes

- Maximize the bandwidth of CPU-to-GPU communication
- The no. of PCIe lanes is limited by the CPU & the motherboard chipset
- Check the block diagram of motherboard chipset

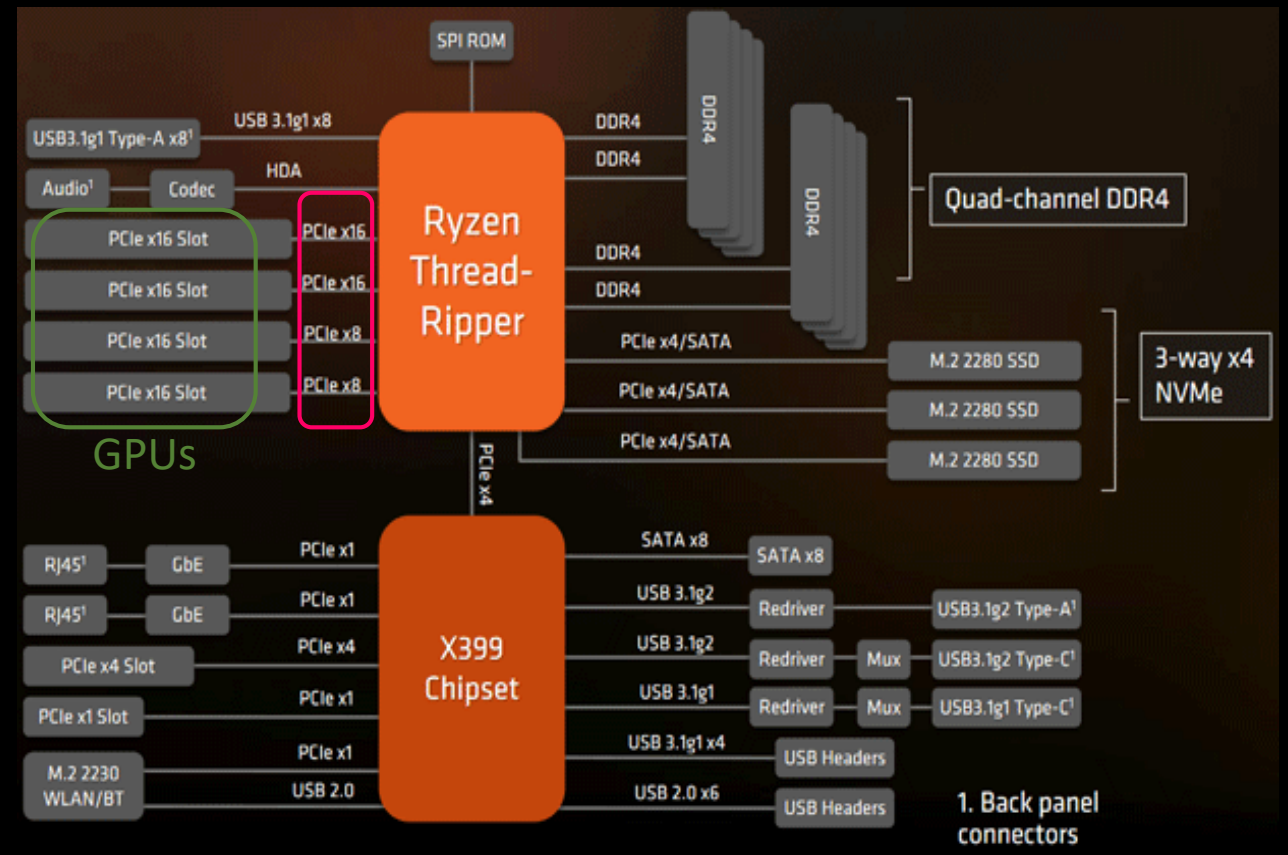


4 PCIe x16 slots ≠ 4 PCIe x16 lanes

Intel

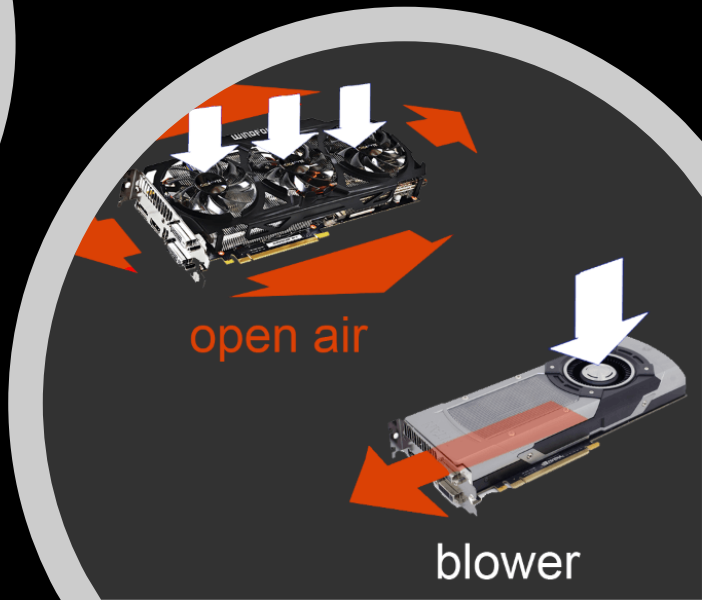


AMD

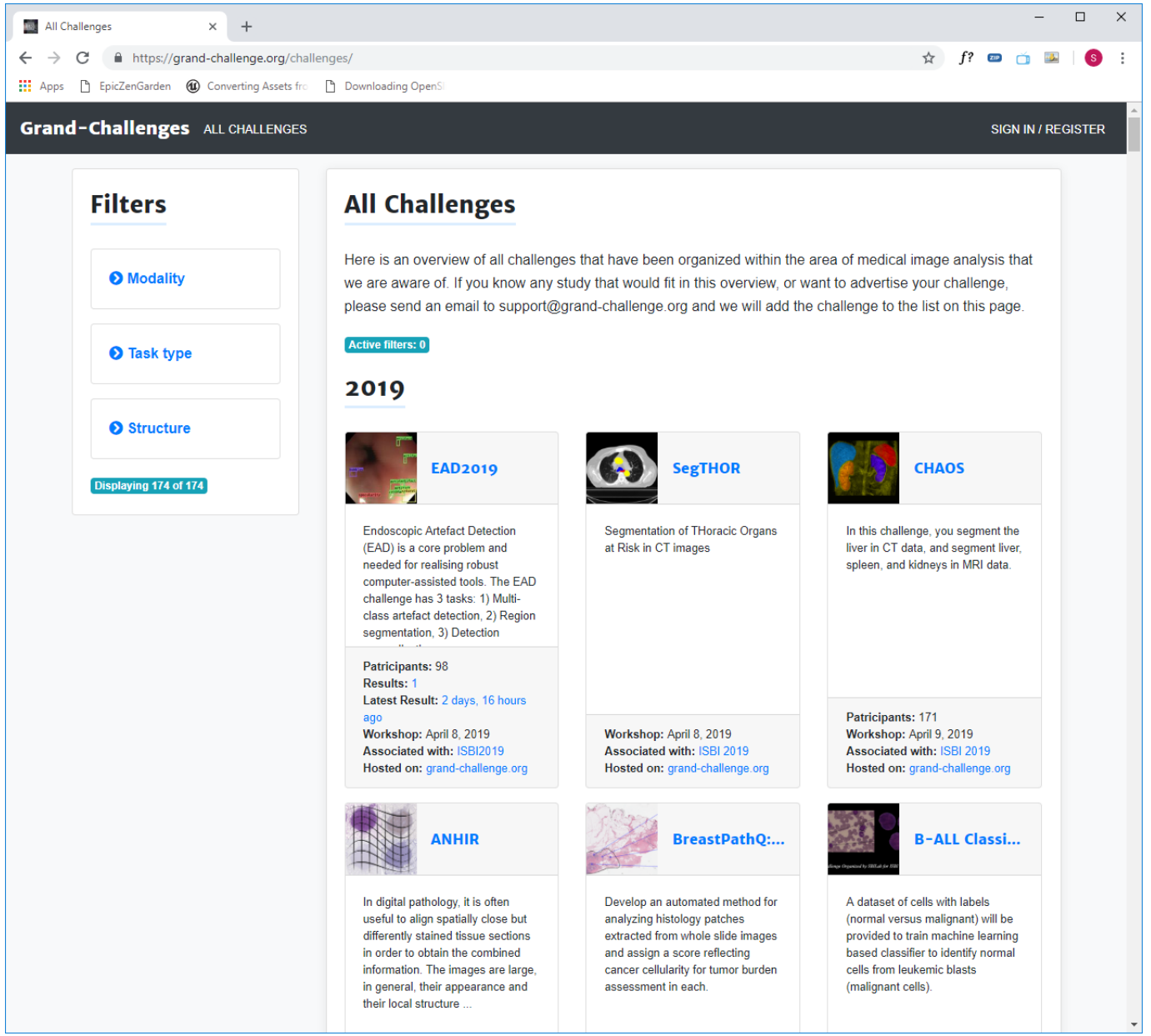


Types of GPU

- Open-air vs Blower
- If you have a spacious case with good air flow and only install 1 GPU, then open air is perfectly fine.
- But if you will stack 4 GPUs into one build, then blower is a wise choice.
- If you have room for an extra fan (and extra $), consider the hybrid
  - It remain at low temperature after prolonged 100% utilization. I have no complaints ….



What’s next ?


All Challenges
https://grand-challenge.org/challenges/
Apps
EpicZenGarden
Converting Assets fro
Downloading OpenSl
Grand-Challenges
ALL CHALLENGES
SIGN IN / REGISTER
Filters
Modality
Task type
Structure
Displaying 174 of 174
All Challenges
Here is an overview of all challenges that have been organized within the area of medical image analysis that we are aware of. If you know any study that would fit in this overview, or want to advertise your challenge, please send an email to support@grand-challenge.org and we will add the challenge to the list on this page.
Active filters: 0
2019
EAD2019
Endoscopic Artefact Detection (EAD) is a core problem and needed for realising robust computer-assisted tools. The EAD challenge has 3 tasks: 1) Multi-class artefact detection, 2) Region segmentation, 3) Detection
Patricipants: 98
Results: 1
Latest Result: 2 days, 16 hours ago
Workshop: April 8, 2019
Associated with: ISBI2019
Hosted on: grand-challenge.org
SegTHOR
Segmentation of THoracic Organs at Risk in CT images
Workshop: April 8, 2019
Associated with: ISBI 2019
Hosted on: grand-challenge.org
CHAOS
In this challenge, you segment the liver in CT data, and segment liver, spleen, and kidneys in MRI data.
Patricipants: 171
Workshop: April 9, 2019
Associated with: ISBI 2019
Hosted on: grand-challenge.org
ANHIR
In digital pathology, it is often useful to align spatially close but differently stained tissue sections in order to obtain the combined information. The images are large, in general, their appearance and their local structure ...
BreastPathQ:...
Develop an automated method for analyzing histology patches extracted from whole slide images and assign a score reflecting cancer cellularity for tumor burden assessment in each.
B-ALL Classi...
A dataset of cells with labels (normal versus malignant) will be provided to train machine learning based classifier to identify normal cells from leukemic blasts (malignant cells).

Thank you!

scottykwok@gmail.com

Slides: github.com/scottykwok/bach2018/

PYCON HK 2018